

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



## Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO FIN DE GRADO

# Monitorización, sincronización y detección de eventos simultáneos de señalización electrofisiológica y comportamiento en vídeo

Autor: Sofía Cerdá Muñoz  
Tutor: Pablo Varona Martínez

Junio 2017



# Monitorización, sincronización y detección de eventos simultáneos de señalización electrofisiológica y comportamiento en vídeo

Autor: Sofía Cerdá Muñoz  
Tutor: Pablo Varona Martínez

Grupo de neurocomputación biológica (GNB)  
Dpto. de Ingeniería informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Junio 2017





## Resumen

La monitorización simultánea y sincronizada del comportamiento de una entidad biológica y el registro de las señales electrofisiológicas que genera su sistema nervioso, ofrece un gran abanico de posibilidades en investigación en el contexto de la biomedicina, la neurociencia y la neuroetología. Sin embargo, muchas veces esto únicamente se puede llevar a cabo recogiendo ambas señales al mismo tiempo, en un dispositivo comercial específico, lo que no siempre es posible.

Por este motivo, en este trabajo se ha decidido diseñar una aplicación que permita sincronizar automáticamente y monitorizar simultáneamente señales de comportamiento en vídeo y señales electrofisiológicas. Con ello se pretende ofrecer flexibilidad a la hora de adquirir los datos y desarrollar una herramienta que funcione de forma genérica para distintos tipos de experimentos que necesitan monitorizar estas señales.

En este trabajo de fin de grado se documenta el diseño y la creación de esta herramienta. Se trata de una aplicación implementada en el lenguaje C++, usando también librerías de OpenCV y Qt para el sistema operativo Linux. Permite visualizar simultáneamente el vídeo y las señales electrofisiológicas, desplazarse tanto por la serie temporal como por los distintos fotogramas, y se adapta a las distintas frecuencias de adquisición de ambas señales.

Para llevar a cabo la sincronización, se debe detectar un artefacto generado simultáneamente en las dos señales, y comenzar la visualización de ambas a partir de este momento. En este trabajo se ha utilizado la activación de un puntero láser como artefacto. El algoritmo de detección del artefacto en el vídeo se basa en la detección del color en cada fotograma, y en la señal electrofisiológica, en la superación de un cierto umbral en la señal que recoge la caída de voltaje de la batería del puntero láser. Este registro se realiza de forma simultánea y en el mismo sistema de adquisición de la señal electrofisiológica.

La herramienta se ha validado con registros de la monitorización de un pez eléctrico (*Gnathonemus petersii*), junto con la actividad eléctrica que éste genera. También se ha llevado a cabo una prueba con la actividad de una neurona electrónica, al tiempo que se recogía en vídeo la representación de la señal en un osciloscopio.

## Palabras Clave

Monitorización simultánea de vídeo y electrofisiología, sincronización de vídeo y series temporales, neuroetología, detección de eventos.

## **Abstract**

Simultaneous and synchronized monitoring of the behavior of a biological entity and the recording of the electrophysiological signals generated by its nervous system, offers a wide range of possibilities in research in the context of biomedicine, neuroscience and neuroethology. However, often this can only be done by collecting both signals at the same time, in a specific commercial device, which is not always possible.

For this reason, in this work we deal with the design of an application that allows automatic synchronization and simultaneous monitoring of behavioral signals in video and electrophysiological signals. This is intended to offer flexibility in acquiring the data and to develop a tool that works in a generic way for different types of experiments that need to monitor these signals.

This final degree project documents the design and creation of this tool. It is an application implemented in the C++ language, also using OpenCV libraries and Qt for the Linux operating system. It allows simultaneous visualization of the video and the electrophysiological signals, shift both through the time series and the different frames, and adapts to the different acquisition frequencies of both signals.

In order to carry out the synchronization, an artifact generated simultaneously in the two signals must be detected, and the visualization of both must begin at this moment. In this work the activation of a laser pointer has been used as artifact. The detection algorithm of the video artifact is based on the detection of the color in each frame and in the electrophysiological signal, in the exceedance of a certain threshold in the signal that collects the voltage drop of the battery of the laser pointer. This record is made simultaneously and in the same acquisition system of the electrophysiological signal.

The tool has been validated with records of the monitoring of an electric fish (*Gnathonemus petersii*), together with the electrical activity generated by it. A test was also carried out with the activity of an electronic neuron, while a video recorded the representation of the signal on an oscilloscope.

## **Key words**

Simultaneous monitoring of video and electrophysiology, synchronization of video and time series, neuroethology, event detection.

# Agradecimientos

En primer lugar, quisiera agradecer el apoyo recibido por mi tutor, Pablo Varona Martínez, por su paciencia y atención, y por brindarme la oportunidad de llevar a cabo este proyecto.

Agradecer también a mi gran amigo y compañero David sus conocimientos sobre la teoría del color, que más de una vez me han sacado de un apuro.

Gracias también a mis padres y a todas aquellas personas que me han apoyado en los momentos de más estrés, que me han dado ánimos y fuerzas para alcanzar mis metas.



# Índice general

<b>Índice de Figuras</b>	<b>IX</b>
<b>Índice de Tablas</b>	<b>XI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación del proyecto . . . . .	1
1.2. Objetivos y enfoque . . . . .	2
1.3. Organización de la memoria . . . . .	3
<b>2. Estado del arte</b>	<b>5</b>
2.1. Bonsai . . . . .	5
2.1.1. Descripción y objetivos de funcionalidad . . . . .	5
2.1.2. Análisis de cumplimiento de requerimientos . . . . .	6
2.2. Ephus . . . . .	7
2.2.1. Descripción y objetivos de funcionalidad . . . . .	7
2.2.2. Análisis de cumplimiento de requerimientos . . . . .	7
2.3. WinEEG . . . . .	8
2.3.1. Descripción y objetivos de funcionalidad . . . . .	8
2.3.2. Análisis de cumplimiento de requerimientos . . . . .	8
2.4. Aplicaciones desarrolladas para experimentos específicos . . . . .	8
2.4.1. Experimento con primates . . . . .	8
2.4.2. Experimento con ratas . . . . .	9
<b>3. Diseño y Desarrollo</b>	<b>11</b>
3.1. Diseño de partida . . . . .	11
3.2. Sincronización y detección de eventos . . . . .	12
3.2.1. Generación de artefactos simultáneos . . . . .	12
3.2.2. Detección de artefactos en el vídeo . . . . .	12
3.2.3. Detección de artefactos en la señal electrofisiológica . . . . .	15
3.2.4. Sincronización automática . . . . .	16

<b>4. Experimentos realizados y resultados</b>	<b>19</b>
4.1. Monitorización del pez eléctrico <i>Gnathonemus Petersii</i> . . . . .	19
4.1.1. Validación de la detección de artefactos . . . . .	19
4.1.2. Análisis de la sincronización . . . . .	21
4.2. Prueba de sincronización con una neurona electrónica . . . . .	25
4.2.1. Validación de la detección de artefactos . . . . .	26
4.2.2. Análisis de la sincronización . . . . .	27
<b>5. Conclusiones y trabajo futuro</b>	<b>33</b>
<b>Glosario de acrónimos</b>	<b>35</b>
<b>Bibliografía</b>	<b>36</b>

# Índice de Figuras

2.1. Lista de nodos Bonsai (Imagen extraída de [1]). . . . .	6
2.2. Ejemplo de flujo de datos para la segmentación de color y el seguimiento de un péndulo en movimiento (Imagen extraída de [1]). . . . .	7
2.3. Polígonos que definen el contorno de los ojos de los primates (Imagen extraída de [2]). . . . .	9
3.1. Ejemplo de generación de artefactos producido por el encendido del puntero láser. . . . .	12
3.2. Código para la detección de un láser en una señal de vídeo. . . . .	13
3.3. Cono de colores del espacio HSV. . . . .	14
3.4. Campana de Gauss. . . . .	14
3.5. Ejemplo de selección de pesos en una máscara gaussiana. . . . .	14
3.6. Umbral para la detección del artefacto en la señal electrofisiológica . . . . .	15
3.8. Código para la sincronización automática de las señales. . . . .	16
3.7. Código para la detección de una diferencia de potencial. . . . .	18
4.1. Datos correspondientes a los artefactos detectados en la prueba del pez eléctrico (a la izquierda la señal electrofisiológica y a la derecha la señal de vídeo). . . . .	20
4.2. Ejemplos de fotogramas en los que se detectó la luz del láser (prueba del pez eléctrico). . . . .	20
4.3. Primeros eventos de la señal que recoge el voltaje del láser (prueba del pez eléctrico). . . . .	20
4.4. Primer artefacto visualizado en el vídeo, pero no en la señal electrofisiológica (prueba del pez eléctrico). . . . .	21
4.5. Primer artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba del pez eléctrico). . . . .	22
4.6. Primer artefacto visualizado en la señal que registra el voltaje del láser (prueba del pez eléctrico). . . . .	22
4.7. Último artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba del pez eléctrico). . . . .	23
4.8. Último artefacto visualizado en la señal que registra el voltaje del láser (prueba del pez eléctrico). . . . .	23
4.9. Datos correspondientes a los artefactos detectados en la prueba de la neurona electrónica (a la izquierda la señal electrofisiológica y a la derecha la señal de vídeo). . . . .	26

4.10. Ejemplos de fotogramas en los que se detectó la luz del láser (prueba de la neurona electrónica). . . . .	26
4.11. Primeros eventos de la señal que recoge el voltaje del láser (prueba de la neurona electrónica). . . . .	27
4.12. Fotograma 1 (izquierda) y 600 (derecha) de la señal de vídeo. . . . .	27
4.13. Primer artefacto visualizado en el vídeo, pero no en la señal electrofisiológica (prueba de la neurona electrónica). . . . .	28
4.14. Segundo artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba de la neurona electrónica). . . . .	28
4.15. Segundo artefacto visualizado en la señal que registra el voltaje del láser (prueba de la neurona electrónica). . . . .	29
4.16. Último artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba de la neurona electrónica). . . . .	29
4.17. Último artefacto visualizado en la señal que registra el voltaje del láser (prueba de la neurona electrónica). . . . .	29



## Índice de Tablas

4.1. Tiempos originales y relativos al tiempo del evento de la sincronización (prueba del pez eléctrico). . . . .	24
4.2. Fotogramas asociados a tiempos de la señal electrofisiológica de los eventos (prueba del pez eléctrico). . . . .	24
4.3. Tiempos asociados a fotogramas de la señal de vídeo de los eventos (prueba del pez eléctrico). . . . .	25
4.4. Tiempos originales y relativos al tiempo del evento de la sincronización (prueba de la neurona electrónica). . . . .	30
4.5. Fotogramas asociados a tiempos de la señal electrofisiológica de los eventos (prueba de la neurona electrónica). . . . .	30
4.6. Tiempos asociados a fotogramas de la señal de vídeo de los eventos (prueba de la neurona electrónica). . . . .	31



# 1

## Introducción

### 1.1. Motivación del proyecto

---

La monitorización simultánea y sincronizada del comportamiento de una entidad biológica y el registro de las señales electrofisiológicas que genera su sistema nervioso, ofrece un gran abanico de posibilidades en investigación, en el contexto de la biomedicina, la neurociencia y la neuroetología [3], [4], [5]. El continuo desarrollo y la imparable mejora de las tecnologías relacionadas con imagen, desemboca en reproductores de vídeo cada vez más rápidos, precisos y de mejor calidad, lo que empuja a llevar a cabo este tipo de experimentos en las distintas ramas mencionadas.

Sin embargo, muchas veces esto únicamente se puede llevar a cabo recogiendo ambas señales al mismo tiempo, en un dispositivo comercial específico. Estas herramientas suelen requerir un desembolso considerable, además de limitar la libertad en la adquisición de los datos, la portabilidad de estos y la interoperabilidad con otros dispositivos. A menudo están diseñadas para un ámbito muy específico que no permite su uso en otros contextos de experimentación.

Por este motivo, surge la necesidad de crear herramientas que permitan la visualización de estas señales de manera sincronizada, sin importar de donde provengan los datos ni el ámbito experimental del estudio. Este proyecto trata de realizar una monitorización simultánea y sincronización automática de una señal de vídeo y una señal electrofisiológica, independientemente del dispositivo con el que se hayan recogido, y la frecuencia de adquisición de cada una de ellas.

Existen diversas líneas de investigación, dentro de los ámbitos mencionados anteriormente, para las que podría ser muy útil esta herramienta. A continuación comentamos algunas de ellas.

En biomedicina, esta aplicación sería de inmensa utilidad para el cuidado y tratamiento de pacientes epilépticos. Al monitorizar al sujeto mediante un electroencefalograma (EEG), registrando su actividad cerebral, al mismo tiempo que se recoge en vídeo su comportamiento físico, se puede llevar a cabo un estudio de la correlación entre la actividad física y neuronal del paciente antes, durante y después de sufrir una crisis. Con ello, se podrían detectar los eventos, físicos o neurológicos, que preceden a estos ataques [6].

En el Grupo de Neurocomputación Biológica (GNB) de la Universidad Autónoma de Madrid (UAM), se desarrollan experimentos para los cuáles esta herramienta sería de gran ayuda en el estudio de los resultados finales. En el ámbito de la neurociencia, el GNB está desarrollando

una interfaz cerebro-máquina que monitoriza la actividad ocular de un individuo, con el fin de detectar los artefactos que se producen con el movimiento de los ojos o el parpadeo. Si al mismo tiempo se recogen en vídeo estas acciones, con ayuda de la aplicación descrita se podría localizar el foco de observación en cada instante, y caracterizarlo con su señal electrofisiológica correspondiente. Esta investigación puede resultar de utilidad para la creación de aplicaciones y dispositivos personalizados y adaptados al usuario, así como para la detección de déficit de atención o dislexia, ya que se puede estudiar el método de lectura de una persona [7], [8], [9].

Dentro del ámbito de la neuroetología, el GNB se ha centrado en la especie *Gnathonemus petersii*, comúnmente conocido como pez elefante, ya que presenta una peculiaridad muy interesante, la electrolocalización. Esta propiedad le permite comunicarse con otros individuos de su especie, orientarse en entornos de visibilidad reducida y defenderse de otros animales. Esto lo consigue mediante la emisión y recepción de impulsos eléctricos, de la misma manera que funciona un sónar. En los experimentos llevado a cabo con este animal, se han recogido mediciones de su actividad eléctrica y una grabación en vídeo de su comportamiento. Con la aplicación descrita, se podría correlacionar y caracterizar los movimientos del pez con los impulsos eléctricos que genera en cada momento. [10], [11], [12].

## **1.2. Objetivos y enfoque**

---

En este trabajo se propone diseñar una herramienta que permita llevar a cabo una monitorización simultánea de señales electrofisiológicas y comportamiento en vídeo, cuyos datos pueden ser recogidos por fuentes distintas. Con esta monitorización, se podría estudiar la correlación de un determinado comportamiento con su correspondiente respuesta electrofisiológica y viceversa.

En primer lugar, se debe realizar una sincronización automática de las señales, ya que al ser adquiridas por dispositivos distintos, es muy posible que exista un cierto retardo entre una señal y otra. Para ello, es necesario generar un artefacto que aparezca en ambas señales al mismo tiempo, con el fin de tener una referencia temporal clara en la que las dos señales coincidan. Este artefacto se debe elegir cuidadosamente para que sea apreciable tanto en la señal de vídeo como en la señal electrofisiológica. En este proyecto se ha elegido como artefacto la luz generada por un puntero láser, y la caída de voltaje producida al activarse el dispositivo. Tras generar el artefacto, se deben aplicar los algoritmos adecuados para su detección en cada una de las señales.

Una vez conocido el momento en el que el artefacto se ha producido en cada señal, se procede a reproducir cada una de ellas a partir de ese instante. Dado que las frecuencias de muestreo son considerablemente distintas, siendo mucho mayor la de la señal electrofisiológica, es necesario analizarlas y adaptarlas para la correcta sincronización y visualización de ambas señales simultáneamente. El hecho de que las frecuencias de muestreo sean tan diferentes puede provocar cierta pérdida de precisión en la sincronización, incrementándose cuanto mayor sea esta diferencia.

Para validar la herramienta se propone realizar una monitorización a un pez elefante (*Gnathonemus petersii*), que, como ya se ha dicho, genera actividad eléctrica, el cual se encontraba en una pecera con electrodos. Con ello se pretende estudiar la correlación entre sus movimientos y los impulsos eléctricos que genera. Para poder llevar a cabo la sincronización, se ha proyectado la luz de un puntero láser dentro de la pecera, de manera que la luz se reflejara en el agua.

También se ha recogido la actividad de una neurona electrónica, junto con la grabación en vídeo de su representación en un osciloscopio. En el vídeo también se observa un cronómetro, que en la fase de análisis permitirá comprobar si la frecuencia de muestreo se mantiene estable a lo largo del tiempo. En ambos experimentos, se ha registrado la diferencia de voltaje producida

por el puntero láser al ser activado, al mismo tiempo que la señal electrofisiológica. Finalmente se hará un análisis de la precisión de la sincronización para cada una de las pruebas.

Por otro lado, es importante la usabilidad de la aplicación, para lo cual presenta una interfaz de usuario sencilla y fácil de entender. Del mismo modo, la aplicación permite desplazarse tanto en la señal de vídeo como a lo largo de la señal temporal durante la visualización simultánea de las señales.

### 1.3. Organización de la memoria

---

La memoria consta de los siguientes capítulos:

- **Introducción:** En esta sección se expondrán las motivaciones por las que se ha llevado a cabo este proyecto, algunos campos de investigación en los que se podría aplicar y una descripción detallada de los objetivos que se pretenden alcanzar.
- **Estado del arte:** En este apartado se mostrarán algunas herramientas con funcionalidades similares a las que se propone en este proyecto, pero que no cumplen con todos los requisitos que aquí se exigen. Se explicarán tanto sus capacidades como sus limitaciones.
- **Diseño y desarrollo:** En primer lugar, se expondrán brevemente las funcionalidades que presentaba la aplicación de partida. A continuación, la mayor parte de la sección estará ocupada en explicar los algoritmos de detección de artefactos utilizados para cada una de las señales, así como el procedimiento llevado a cabo para realizar la sincronización automática.
- **Experimentos realizados y resultados:** Se detallarán las pruebas realizadas para validar la detección de eventos y la sincronización automática de las señales. También se realizará un análisis de la precisión de la sincronización conseguida en cada caso.
- **Conclusiones y trabajo futuro:** Finalmente, la memoria se cerrará con una conclusión sobre el trabajo realizado y los objetivos conseguidos, así como posibles mejoras que podrían llevarse a cabo en el futuro.



# 2

## Estado del arte

Existen distintas aplicaciones comerciales con funcionalidades similares a las definidas para este trabajo en contextos de investigación específicos, y también en la mayoría de los casos para dispositivos concretos. En esta sección se expondrán algunos de estos programas, explicando sus objetivos y funcionalidades. También se realizará un análisis del cumplimiento de los requerimientos definidos en la sección anterior. Finalmente se mencionarán algunas investigaciones, dentro del ámbito de la neuroetología, para las que se han desarrollado herramientas con funcionalidades similares a las buscadas en este proyecto, pero específicas para cada tipo de experimento.

### 2.1. Bonsai








---

#### 2.1.1. Descripción y objetivos de funcionalidad

El objetivo principal de Bonsai es simplificar y acelerar el desarrollo de software para la adquisición y procesamiento de las muchas fuentes de datos heterogéneas utilizadas comúnmente en investigación científica (concretamente en neurociencia). Es un framework de uso genérico basado en eventos para el procesamiento y control de flujo de datos. Permite la creación rápida y flexible de prototipos para el diseño de experimentos integrados en neurociencia, lo que ayuda a reducir el tiempo dedicado a la creación del experimento. Para alcanzar ese objetivo, permite desarrollar software que pueda manejar datos en paralelo y de forma asíncrona (que es como se adquieren normalmente los datos científicos) y es capaz de integrar componentes software y hardware de los distintos dispositivos que participan en un experimento. Además, es de código abierto, modular y de alto rendimiento [1].

El flujo de datos se representa con un grafo cuyos nodos (ver figura 2.1) son secuencias observables (p.ej: un vídeo), que se pueden clasificar como fuentes (*sources*) o como combinadores (*combinators*). Las fuentes proporcionan acceso a datos sin procesar, como imágenes de una cámara de vídeo u ondas de señal de un micrófono o de un amplificador de electrofisiología. Los combinadores representan cualquier operador observable que maneje una o más de estas secuencias. Existen distintos tipos dentro de esta categoría según la manera en la que manipulan sus entradas, como, por ejemplo, las transformaciones (*transforms*) y las llamadas *sinks* (sumideros). Las transformaciones modifican los elementos de datos entrantes de una sola secuencia

de entrada. Por otro lado, las *sinks* manipulan los elementos de entrada pero sin modificar en absoluto la secuencia original, es decir, los cambios se reflejan en otra secuencia.

Color	Category	Description
<b>BONSAI NODE CATEGORIES</b>		
	Source	<b># Inputs: 0</b> Generates observable sequences of data
	Combinator	<b># Inputs: 0..N</b> Can change both the elements and the sequence; elements may be dropped, shifted, duplicated or merged with other sequences
	Transform	<b># Inputs: 1</b> Transforms the elements in the sequence but does not change the sequence order or timing
	Condition	<b># Inputs: 1</b> Does not change the elements in the sequence but can drop (filter out) elements from the sequence
	Sink	<b># Inputs: 1</b> Does not change neither the elements nor the sequence, but can introduce side-effects
	Property	<b># Inputs: 1</b> Represents a property of another node. Writing to it at runtime will change the value of the property
	Nested	<b># Inputs: 0..N</b> A nested combinator uses an internal (nested) dataflow to implement specific operations

*The color of each Bonsai node serves as a visual aid to identify their role in dataflow processing pipelines. Most of these categories are actually specializations of the very general combinator and are meant to visually depict their specific data processing semantics.*

Figura 2.1: Lista de nodos Bonsai (Imagen extraída de [1]).

### 2.1.2. Análisis de cumplimiento de requerimientos

Como podemos observar en la figura 2.2, Bonsai permite visualizar datos procedentes de fuentes muy diversas, además de procesar y controlar el flujo de datos. En nuestro caso, podríamos visualizar la señal de vídeo y la señal electrofisiológica.

Sin embargo, presenta dificultades a la hora de trabajar con procesos que se ejecutan de forma simultánea cuyos flujos de datos se registran de forma paralela en dispositivos distintos, es decir, no realiza labores de sincronización de procesos. Por otro lado, sólo está soportado para el sistema operativo Windows (Windows 7 o superiores).



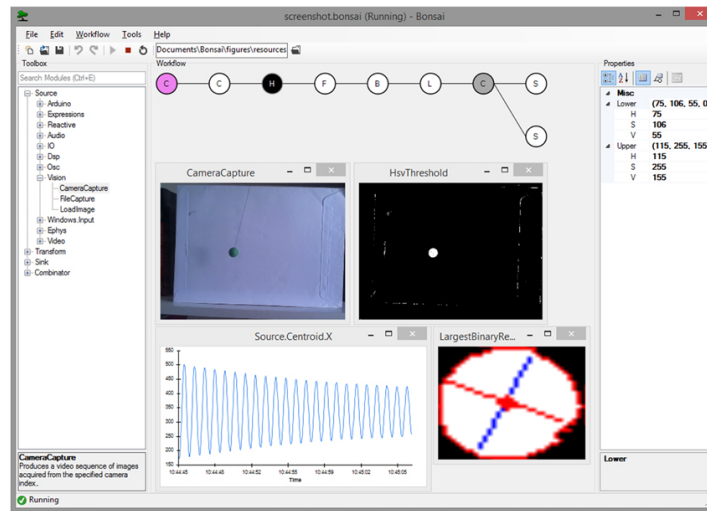


Figura 2.2: Ejemplo de flujo de datos para la segmentación de color y el seguimiento de un péndulo en movimiento (Imagen extraída de [1]).

## 2.2. Ephus

### 2.2.1. Descripción y objetivos de funcionalidad

Ephus es un software de código abierto para la adquisición de datos multipropósito para experimentos en neurociencia [13]. Destaca por su versatilidad, ya que las funciones de usuario son personalizables. Esto se incluye tanto en el análisis en línea como en experimentos en bucle cerrado, donde los parámetros experimentales se pueden cambiar basándose en datos adquiridos recientemente o durante experimentos de comportamiento en tiempo real. También es compatible con varias herramientas para la adquisición de datos y hardware de imágenes.

Este paquete está basado en módulos e incluye funcionalidad para una amplia variedad de aplicaciones que van desde la electrofisiología in vitro tradicional hasta la cartografía altamente personalizada de circuitos y protocolos de comportamiento en tiempo real. Ofrece una colección de herramientas específicas de aplicación, como las de electrofisiología, exploración y mapeo, imágenes de vídeo, así como herramientas comunes para la vinculación de datos y sincronización experimental.

Ephus se ejecuta dentro de Matlab (versión 2007b y superior) como una serie de scripts, clases y cajas de herramientas personalizadas. El usuario puede controlar programas individuales directamente. Además, cada programa puede ser activado por otros programas, o él mismo puede disparar otros programas cuando se inicia.

### 2.2.2. Análisis de cumplimiento de requerimientos

Permite la visualización de un vídeo y una señal electrofisiológica de manera simultánea, pero no la sincronización de ambos si estos se han adquirido por separado o en distinto momento (es decir, que una señal comenzara a registrarse antes que la otra).

Como en el caso de Bonsai, este software está dirigido a facilitar el desarrollo de experimentos en neurociencia, lo cual se aleja del objetivo de este proyecto, que está más relacionado con el análisis de eventos simultáneos de vídeo y electrofisiología.

## **2.3. WinEEG**

---

### **2.3.1. Descripción y objetivos de funcionalidad**

WinEEG, en su versión básica, es un programa que permite la grabación, edición y análisis de EEG grabado de forma continua, usando para ello un hardware específico, un amplificador Mitsar (<http://www.mitsar-medical.com/eeg-software/qeeg-software/>). Sin embargo, la versión avanzada permite al usuario importar ficheros grabados en otros formatos de los creados por los amplificadores Mitsar.

La interfaz gráfica permite mostrar hasta 32 canales de EEG de manera simultánea, junto con la señal de vídeo en la que se graba el comportamiento del sujeto. Con el uso de ciertas funciones y la versión avanzada de WinEEG, el EEG se puede grabar de forma sincronizada con vídeo de una o dos cámaras y señal de audio del micrófono. Además, permite el posicionamiento rápido a cualquier momento seleccionado del vídeo y la detección automática de picos.

### **2.3.2. Análisis de cumplimiento de requerimientos**

Al igual que en los apartados anteriores, este dispositivo permite visualizar simultáneamente vídeo y señales electrofisiológicas, lo que cumple uno de los objetivos buscados. Sin embargo, el programa está diseñado para ser integrado con un hardware que adquiere de manera síncrona todas las señales, pero no es capaz de sincronizar las señales si han empezado a grabar en distintos momentos del tiempo. Por otro lado, su licencia de uso es privada.

## **2.4. Aplicaciones desarrolladas para experimentos específicos**

---

En muchos casos los laboratorios desarrollan aplicaciones para llevar a cabo experimentos concretos, debido a que no existe una herramienta genérica lo suficientemente flexible para su uso particular. A continuación se mencionarán dos experimentos para los que se desarrolló una aplicación específica y exclusiva.

### **2.4.1. Experimento con primates**

Se ha realizado un estudio en el que se monitoriza a primates mientras duermen, con el objetivo de detectar momentos de desvelo. Se recoge la actividad ocular del animal mediante una grabación en vídeo junto con una señal electrofisiológica que muestra la actividad cerebral [2]. Para ello el primate lleva un implante.

Las imágenes de la cara del animal se obtuvieron con una cámara USB que se colocó en una posición fija, a 20 cm de distancia de la cara del animal. El vídeo se recogió a 17 fps (fotogramas por segundo), pero para analizar mejor el parpadeo del primate, esta tasa se redujo a 1 fps. Para llevar a cabo el análisis, se marcaron los contornos de los ojos con polígonos (ver figura 2.3).

Las señales de EEG se filtraron, entre 1 y 70 Hz, y se amplificaron ( $\times 10.000$ ). Para un análisis posterior, la señal de EEG se filtró digitalmente con una banda de paso de 1-13 Hz. Las grabaciones de vídeo y EEG se realizaron simultáneamente y se almacenaron en el mismo ordenador, sincronizadas con el reloj de la PC.

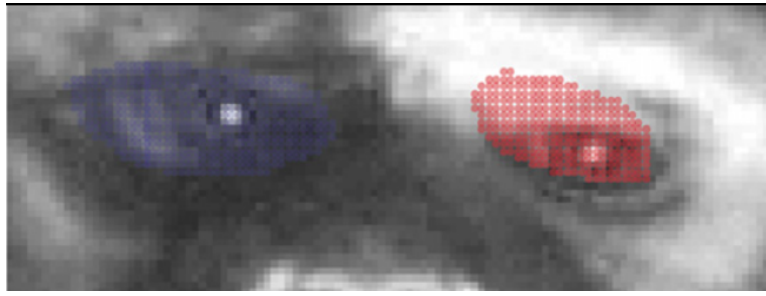


Figura 2.3: Polígonos que definen el contorno de los ojos de los primates (Imagen extraída de [2]).

#### **2.4.2. Experimento con ratas**

Se diseñó un ordenador que registraba en tiempo real tanto una grabación en vídeo como la actividad electrofisiológica de una rata de manera simultánea. Este dispositivo se utilizó para monitorizar el comportamiento de ratas y su movimiento [14].

Al igual que en el apartado anterior, las ratas llevaban un implante que recoge la actividad cerebral del animal. También llevaban en la cabeza dos luces LED de color blanco, para poder localizar mejor el lugar exacto en el que se encontraban en cada momento. Para detectar estas luces, se aplicaba un filtro a cada fotograma que detectaba los puntos de la imagen que presentaban un brillo por encima de un umbral fijado.



# 3

## Diseño y Desarrollo

En esta sección, primeramente se hará un breve resumen de las funcionalidades que presentaba la aplicación de partida, y después se expondrán los distintos algoritmos utilizados para realizar la sincronización automática y la visualización simultánea de las señales.

### 3.1. Diseño de partida

---

Este proyecto partía de una aplicación ya desarrollada que permitía visualizar una señal de vídeo y una señal electrofisiológica simultáneamente. En ella se implementaron las funciones que permitían cargar ambas señales y visualizarlas, así como gestionar la interfaz gráfica de la aplicación. También presentaba funcionalidades de navegación durante la reproducción, como la opción de pausar o resetear al inicio de los registros [15].

Sin embargo, la sincronización de las señales se debía hacer de modo manual, navegando hasta el fotograma del vídeo en el que se encontraba el artefacto, para conocer el instante de sincronización. Esta sincronización manual sirvió de orientación para llevar a cabo la sincronización automática, pero fue necesario realizar múltiples cambios y mejoras en toda la implementación de la aplicación para poder llevar a término esta modificación.

Para llevar a cabo la implementación de la herramienta, se emplearon las librerías de OpenCV, el framework multiplataforma Qt y el lenguaje de programación de alto nivel C++. Las librerías de OpenCV están especializadas en la gestión, análisis y procesamiento de imágenes en tiempo real, por lo que se utilizaban para llevar a cabo la visualización del vídeo durante la monitorización. Con las funciones de Qt se gestionaba toda la interfaz gráfica, además de servir como entorno de programación. Tanto las librerías de OpenCV como las de Qt se encontraban implementadas en Python y C++, pero se decidió utilizar este último lenguaje debido a su mayor potencia.

Para este proyecto, se mantuvo la elección de las librerías y el lenguaje de programación. Se extendió el uso de las librerías de OpenCV para aplicar transformaciones a los fotogramas, con el fin de detectar el artefacto deseado.

## 3.2. Sincronización y detección de eventos

---

### 3.2.1. Generación de artefactos simultáneos

Para llevar a cabo la sincronización automática es imprescindible que aparezca un artefacto que se genere al mismo tiempo en la señal electrofisiológica y en la de vídeo. De esta manera, aunque ambas señales se hayan adquirido en momentos distintos de tiempo (en el sentido de que una haya empezado antes que otra), al llevar a cabo la sincronización, el análisis de la correlación de eventos en las dos modalidades podrá comenzar en el momento en el que se detectó el artefacto en las dos señales. A partir de este punto, si se adaptan correctamente las frecuencias de cada señal para su visualización simultánea (que se explicará en la sección 3.2.4 en la ecuación 3.1), estas se reproducirán de forma sincronizada, ya que ahora sí comienzan al mismo tiempo.

En este proyecto, se propuso que el artefacto generado proviniera de un puntero láser, cuyos polos positivo y negativo de su batería se encontraban conectados a un canal de la tarjeta de adquisición de datos de la señal electrofisiológica. De esta forma, al presionar el interruptor se registraba la caída de voltaje que producía la pérdida de carga, lo que se representa como una diferencia de potencial. Debemos matizar también, que el voltaje del láser se registra en un canal y la señal electrofisiológica de origen biológico en otro simultáneamente, con lo que se conseguía que ésta última no se viera afectada por los artefactos del canal del láser.

Con ello, se conseguía generar un artefacto en el vídeo al iluminar con el láser el entorno que se pretendía grabar, y al mismo tiempo, se recogía el artefacto producido por la diferencia de potencial, en el momento exacto de la activación de la luz. Se decidió emplear este tipo de artefacto dado que se trataba de una señalización óptica fácil de detectar, por su luminosidad y forma reconocibles. Por otro lado, este es un dispositivo que está disponible en cualquier laboratorio de investigación a un coste reducido. En la figura 3.1, se puede observar una gráfica que muestra la diferencia de potencial descrita.

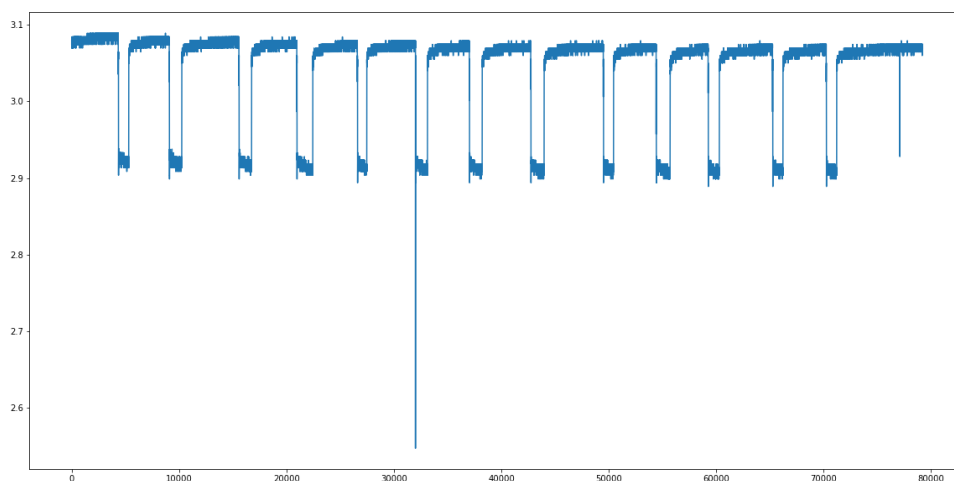


Figura 3.1: Ejemplo de generación de artefactos producido por el encendido del puntero láser.

### 3.2.2. Detección de artefactos en el vídeo

Para llevar a cabo la detección de la luz del puntero láser en la señal de vídeo, se han utilizado funciones proporcionadas por la librería OpenCV. Las transformaciones realizadas a los fotogramas para detectar en cuáles de ellos aparecía la luz del láser, han consistido principalmente en aplicar un filtro de color con el fin de captar únicamente el color del láser. También se podría

haber optado por detectar la forma circular del láser, pero realizando pruebas se optó finalmente por la primera opción por su simplicidad y eficacia. En la figura 3.2 se muestra el código que realiza esta tarea.

```
int AutoSync::detectarLaserVideo.VideoCapture vc, int umbral_noCeros,
                        int hmin, int smin, int vmin, int hmax, int smax, int vmax) {
    Mat matOriginal, matOriginalHSV, matProcessed;

    int num_totalFrames = vc.get(CV_CAP_PROP_FRAME_COUNT);
    int num_frameLeido = vc.get(CV_CAP_PROP_POS_FRAMES);
    int noCeros, numEventos = 0, ultimo_frameEvento = -2;

    QFile fichero_salida("../eventos/eventos_video.txt");
    if (!fichero_salida.open(QIODevice::WriteOnly)) {
        cout << "ERROR AL ESCRIBIR EN EL FICHERO DE EVENTOS VIDEO" << endl;
        return -1;
    }
    QTextStream tsw(&fichero_salida);

    while (num_frameLeido < num_totalFrames) {
        vc.read(matOriginal);
        num_frameLeido = vc.get(CV_CAP_PROP_POS_FRAMES);

        // converting to HSV
        cvtColor(matOriginal, matOriginalHSV, CV_BGR2HSV);
        // search of laser spot
        inRange(matOriginalHSV, Scalar(hmin, smin, vmin), Scalar(hmax, smax, vmax), matProcessed);
        // blur
        GaussianBlur(matProcessed, matProcessed, Size(9, 9), 2.2);

        noCeros = countNonZero(matProcessed);
        if (noCeros > umbral_noCeros) {
            if (ultimo_frameEvento <= (num_frameLeido-100)) {
                cout << "***** LASER DETECTADO EN FRAME " << num_frameLeido << endl;
                numEventos++;
                tsw << numEventos << " ";
                tsw << num_frameLeido << endl;
            }
            ultimo_frameEvento = num_frameLeido;
        }
    }
    fichero_salida.close();

    return numEventos;
}
```

Figura 3.2: Código para la detección de un láser en una señal de vídeo.

Primeramente, a cada fotograma leído se le aplica un filtro de color, captando únicamente los píxeles con colores que se encuentran dentro de un rango establecido según el formato HSV. Después se aplicaba un suavizado gaussiano para difuminar la imagen, eliminando así el ruido y las aberraciones de color de la mejor manera posible. A continuación se explican con más detalle las funciones utilizadas para realizar estas transformaciones.

Para aplicar el filtro de color se ha empleado el formato HSV (del inglés *Hue*, *Saturation*, *Value* - Matiz, Saturación, Valor), ya que con este modelo es más fácil analizar imágenes. Se trata de una transformación no lineal del espacio de color RGB, que es el más común para representar colores en pantalla. El matiz indica el código de color y se representa en una circunferencia. Por su parte, la saturación hace referencia a la pureza del color, es decir, cuanto menor sea este parámetro más se asemejará a la escala de grises, y se representa como la distancia al eje de brillo negro-blanco. Finalmente, el valor indica la exposición o la luminosidad del color, también llamado brillo, que se representa como la altura en el eje blanco-negro (ver figura 3.3). El matiz toma valores entre 0 y 360 (los grados de una circunferencia), aunque en OpenCV estos valores se miden en porcentaje de 0 a 100, lo que implica hacer una conversión sencilla. El

resto de parámetros normalmente toman valores entre 0 y 255, ya que comúnmente los píxeles de una imagen contienen 8 bits de información, y en este caso, se usa la misma representación en OpenCV.

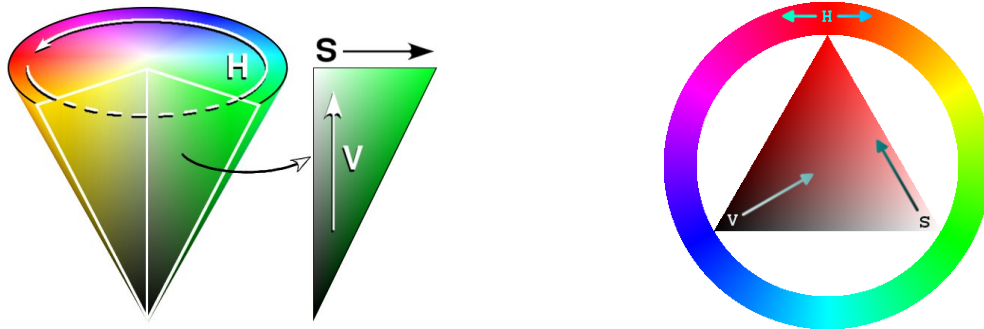


Figura 3.3: Cono de colores del espacio HSV.

La función *GaussianBlur* calcula una media ponderada, donde los pesos de los píxeles toman la forma de una campana de Gauss, con  $f(x) = e^{\frac{-x^2}{s^2}}$  (ver figura 3.4). Estos pesos dependen de la distancia al píxel central de la máscara (ver figura 3.5). La varianza,  $s^2$ , indica el nivel de suavizado, de manera que si tiene un valor grande la campana será más ancha, y por tanto habrá más suavizado, y si el valor es pequeño, la campana será más estrecha y por consiguiente, el suavizado será menor. Este parámetro se pasa como argumento a la función, que nosotros hemos decidido darle el valor 2.2, así como el tamaño del núcleo de Gauss, que hemos decidido que sea *Size(9,9)*.

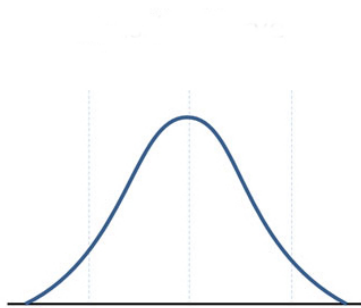


Figura 3.4: Campana de Gauss.

Máscara  
gaussiana de 3x3

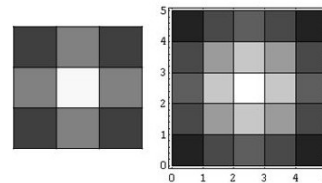
$$\frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$


Figura 3.5: Ejemplo de selección de pesos en una máscara gaussiana.

Tras aplicar todas estas modificaciones descritas, se comprueba en cada fotograma el número de píxeles que aparecen, y cuando esta cantidad supera un umbral establecido, significa que se ha detectado el artefacto. El motivo de realizar la detección de esta manera, tiene su origen en que al aplicar un filtro a la imagen, sólo aparecerán en el fotograma aquellos píxeles que capta dicho filtro, en este caso el color del láser. El umbral citado debería ser 0 si el filtro se ha aplicado correctamente. Sin embargo se ha observado que en ocasiones aparecen aberraciones en el vídeo que generan ruido y que el suavizado gaussiano no consigue eliminar por completo, lo que lleva a la necesidad de establecer un umbral mayor. Finalmente, una vez detectados todos los fotogramas en los que aparece cada artefacto por primera vez, se anotan en un fichero externo.



### 3.2.3. Detección de artefactos en la señal electrofisiológica

Para detectar la diferencia de voltaje producida al activar la luz del puntero láser, se plantearon distintos algoritmos. El primero de ellos, trataba de estudiar los puntos de la señal que superaban cierto umbral en el voltaje, superiormente o inferiormente, dependiendo de cómo estuvieran conectados los polos de la batería del láser a la tarjeta de adquisición de datos. Sin embargo, se planteó la posibilidad de que las señales pudieran presentar deriva. Por ello, la decisión de realizar la detección por umbral no parecía la más adecuada, por lo que se optó por intentar llevar a cabo el estudio de la pendiente de la gráfica y detectar el artefacto cuando dicha pendiente fuera suficientemente grande (en valor absoluto). A pesar de ello, finalmente se volvió a valorar la opción de detectar el artefacto mediante la superación de un umbral, dado que generalmente las señales no presentaban el problema de la deriva y además este algoritmo es mucho más sencillo y fácil de implementar. Una vez más la opción escogida corresponde al balance entre sencillez y eficacia.

Para este método, se ha elegido como umbral el valor representado en la figura 3.6 con una línea horizontal de color naranja. La razón por la que se ha escogido este umbral se debe a que es el valor más cercano a la cota inferior (en este caso) con el cual podemos distinguir el inicio de cada artefacto al leer los datos recogidos. El valor umbral ideal sería precisamente ese "pico", ya que es el momento en el que empieza a aparecer la luz del láser, pero desde ese punto hasta llegar al valor umbral existen oscilaciones que dificultan la detección precisa de cada artefacto (sólo debería superarse el umbral una única vez para cada evento). Esto conlleva una ligera pérdida de precisión en la detección del artefacto, pero ésta era mínima ya que la frecuencia de muestreo de la señal electrofisiológica suele ser muy elevada.

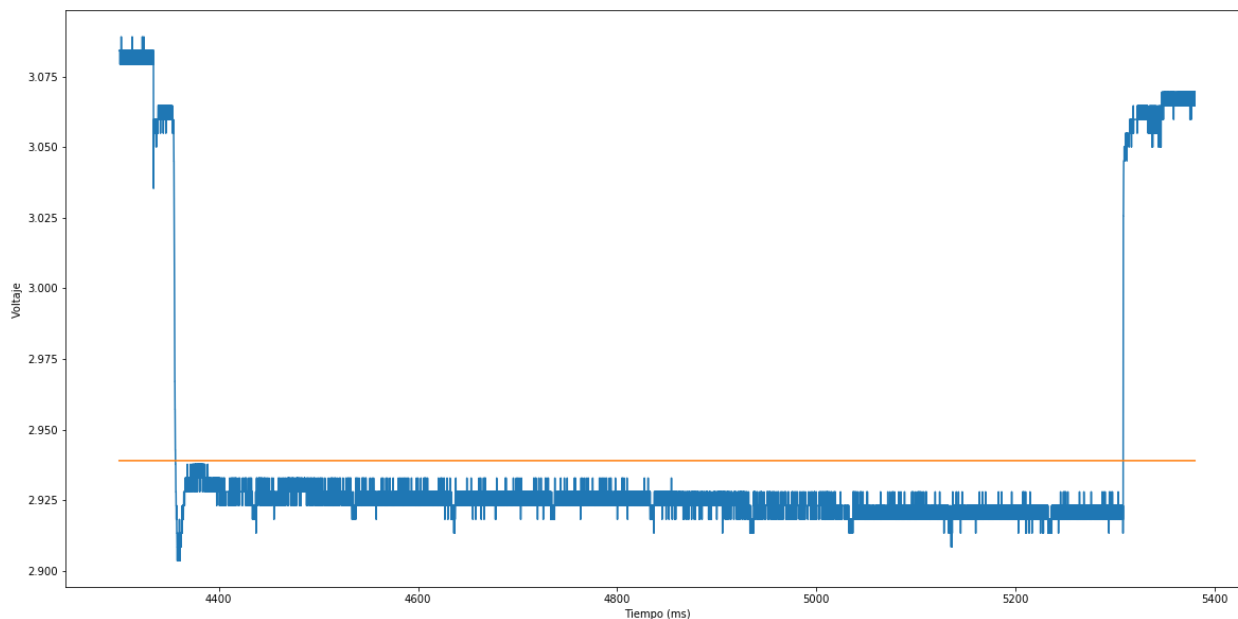


Figura 3.6: Umbral para la detección del artefacto en la señal electrofisiológica

En la figura 3.7 se muestra el código utilizado para dicha tarea. Tras detectar los puntos que superan el umbral establecido (pasado por argumento a la función *detectarLaserBio*), se anotan en un fichero externo el número de muestra de la señal en el que se encuentran estos puntos, el valor del tiempo al que corresponden y el voltaje alcanzado. Únicamente se anota la primera muestra que supera dicho umbral, para lo cual se controla el aumento de la diferencia de potencial (lo que en el código se llama *subida*) que representa el proceso de activación del láser, así como la disminución de ésta cuando el láser se está apagando.

### 3.2.4. Sincronización automática

Para llevar a cabo la sincronización automática hay que hacer corresponder en el tiempo las dos señales teniendo en cuenta que empiezan en instantes de tiempo distintos y tienen frecuencias de muestreo diferentes (en particular la del vídeo con una frecuencia muy baja comparada con la de la señal electrofisiológica). Esa referencia temporal común a ambas señales se conoce tras detectar en cada una de ellas el artefacto generado. En la figura 3.8 se muestra el código que realiza esta tarea, y a continuación explicaremos más en detalle el procedimiento utilizado.

En el código, podemos observar que en primer lugar se resetea la reproducción, y todas las variables implicadas en ello, con la función *on\_stopToolButton\_clicked*, que es la misma que se activa cuando el usuario selecciona el botón *stop* en la interfaz de la aplicación. Esta acción se realiza con el fin de calcular la posición en la que se encuentra el artefacto, partiendo del inicio original de cada señal.

Posteriormente, se procede a leer los archivos en los que se registraron los artefactos detectados en cada señal mediante las funciones *detectarLaserVideo* y *detectarLaserBio*, explicadas anteriormente. En estos ficheros se lee únicamente la primera línea, ya que sólo necesitamos el primer artefacto para realizar la sincronización. La llamada a estas funciones, se lleva a cabo en el momento en el que la aplicación recibe los datos por primera vez, de manera que la detección de los eventos se realiza una única vez y hace la sincronización más rápida y eficiente. Estas mismas funciones, reciben como parámetros los valores obtenidos empíricamente tras realizar las pruebas que se expondrán en la sección 4. Una vez hallados el fotograma y la muestra en la que aparece el primer artefacto, en sus respectivas señales, se toman estos parámetros como referencia temporal inicial ( $t=0$ ) para cada una de las señales.

```
void MainWindow::on_syncToolButton_clicked() {
    int frameDetect, sampleDetect;

    on_stopToolButton_clicked();

    if (sync == 1) {
        QFile eventos_video("../eventos/eventos_video.txt");
        if (eventos_video.open(QIODevice::ReadOnly)) {
            QTextStream tsr(&eventos_video);
            QString linea = tsr.readLine();
            frameDetect = linea.split(" ").at(1).toInt();
            jumpVideo(frameDetect);
            isVideoSync = 1;
        } else {
            cout << "ERROR AL LEER EL FICHERO DE EVENTOS VIDEO" << endl;
            return;
        }

        QFile eventos_bio("../eventos/eventos_bio.txt");
        if (eventos_bio.open(QIODevice::ReadOnly)) {
            QTextStream tsr(&eventos_bio);
            QString linea = tsr.readLine();
            sampleDetect = linea.split(" ").at(1).toInt();
            jumpBioSample(sampleDetect);
            isBioSync = 1;
        } else {
            cout << "ERROR AL LEER EL FICHERO DE EVENTOS BIO" << endl;
            return;
        }
    } else {
        qDebug() << "SINCRONIZACION AUTOMATICA NO DISPONIBLE";
    }
}
```

Figura 3.8: Código para la sincronización automática de las señales.

Para la visualización simultánea ha sido necesario realizar un cálculo para saber cuántas muestras de la serie temporal corresponden a un fotograma de la señal de vídeo, ya que la frecuencia de adquisición es diferente en ambos casos. Éste cálculo se describe en la ecuación 3.1. Esta conversión puede traer consigo una pérdida de precisión durante la sincronización, que se estudiará con detalle en la sección 4. A partir de este resultado, por cada fotograma reproducido se visualizan el número de muestras obtenidas en la conversión.

$$\frac{frecuencia_{senalBio}}{frecuencia_{senalVideo}} = \frac{muestras/segundo}{fotogramas/segundo} = muestras/fotograma \quad (3.1)$$

Finalmente, cabe destacar que la sincronización se lleva a cabo cada vez que se selecciona el botón correspondiente en la interfaz de la aplicación, pero siempre con respecto al inicio de la señal original. Es decir, no se tienen en cuenta los parámetros seleccionados en sincronizaciones anteriores. Además, a la hora de llevar a cabo la monitorización simultánea, en la señal electrofisiológica se muestran los puntos en los que aparece el artefacto, tanto si se está en modo sincronización como si no, para que sea más sencillo comprobar que la sincronización es correcta sin necesidad de mostrar la señal del voltaje del puntero láser (que es con la que se han detectado los artefactos y que se ha registrado al mismo tiempo que la señal electrofisiológica por otro canal de la tarjeta de adquisición de datos).

```
int AutoSync::detectarLaserBio(QString nombre_fichero, double umbral) {
    int num_muestraLeida = 0;
    int numEventos = 0;

    QFile fichero_salida("../eventos/eventos_bio.txt");
    if (!fichero_salida.open(QIODevice::WriteOnly)) {
        cout << "ERROR AL ESCRIBIR EN EL FICHERO DE EVENTOS BIO" << endl;
        return -1;
    }
    QTextStream tsw(&fichero_salida);

    QFile fichero(nombre_fichero);
    if (fichero.open(QIODevice::ReadOnly)) {
        QTextStream tsr(&fichero);
        QString tiempo, amplitud, tiempoEvento, amplitudEvento;
        QStringList lineaList;

        double amplitud_double;
        int subida = 0, muestraEvento;

        while (!tsr.atEnd()) {
            lineaList = tsr.readLine().split(" ");
            if (tsr.status() == QTextStream::Ok) {
                tiempo = lineaList.first();
                amplitud = lineaList.last();

                num_muestraLeida++;

                amplitud_double = amplitud.toDouble();

                if ((subida == 0) && (amplitud_double <= umbral)) {
                    subida = 1;
                    cout << "***** UMBRAL DETECTADO EN MUESTRA " << num_muestraLeida << endl;
                    muestraEvento = num_muestraLeida;
                    tiempoEvento = tiempo;
                    amplitudEvento = amplitud;
                } else if ((subida == 1) && (amplitud_double >= umbral)) {
                    subida = 0;
                    numEventos++;
                    tsw << numEventos << " ";
                    tsw << muestraEvento << " ";
                    tsw << tiempoEvento << " ";
                    tsw << amplitudEvento << endl;
                }
            } else {
                cout << "ERROR AL LEER LA MUESTRA " << num_muestraLeida+1 << endl;
                numEventos = -1;
                break;
            }
        }
        fichero.close();
        fichero_salida.close();
    } else {
        cout << "ERROR AL LEER EL FICHERO BIO" << endl;
        numEventos = -1;
    }

    return numEventos;
}
```

Figura 3.7: Código para la detección de una diferencia de potencial.

# 4

## Experimentos realizados y resultados

Para realizar las pruebas de sincronización, se ha llevado a cabo la monitorización en vídeo del pez eléctrico *Gnathonemus Petersii*, así como el registro de su actividad eléctrica dentro de una pecera con electrodos en su interior.

Además, se ha registrado la actividad de una neurona electrónica conectada a un osciloscopio, al cual también se han conectado los polos de la batería del puntero láser para generar los artefactos al pulsar el botón de encendido. En este caso, se ha añadido un cronómetro para comprobar que el vídeo grababa uniformemente a lo largo del tiempo.

### **4.1. Monitorización del pez eléctrico *Gnathonemus Petersii***

---

En esta prueba se grabó en vídeo el comportamiento del pez durante 1 minuto y 12 segundos, el cual se encontraba dentro de una pecera con electrodos que captaban las señales eléctricas que el animal emitía con su movimiento. La frecuencia de adquisición del vídeo era de 60 fotogramas por segundo, y la señal eléctrica se recogió a 15 KH (es decir, 15000 muestras por segundo). Por tanto, aplicando la fórmula 3.1, obtenemos que se reproducirán 250 muestras de la señal eléctrica por cada fotograma del vídeo.

Como se ha explicado en secciones anteriores, la sincronización se realizó mediante la señalización de un puntero láser proyectado dentro de la pecera. El encendido provocaba una señal eléctrica que se registraba por un segundo canal de la tarjeta de adquisición, de forma simultánea al registro de la actividad del pez eléctrico. Este artefacto se generó hasta en 13 ocasiones, con el fin de comprobar que la sincronización era correcta durante toda la monitorización, y no únicamente para el primer artefacto detectado durante la misma.

#### **4.1.1. Validación de la detección de artefactos**

Al realizar la detección de los artefactos en ambas señales, sus respectivas funciones generaron sendos ficheros en los que se recogieron dichos artefactos y su temporalidad. Las muestras y los fotogramas en los cuales se produjeron los distintos artefactos de esta prueba, se visualizan en la figura 4.1. En el caso de la señal electrofisiológica, en cada línea aparece, en este orden, el número del evento, el número de muestra en el que se detectó el mismo, a continuación el

tiempo en milisegundos correspondiente a esa muestra, y finalmente la diferencia de potencial de la carga del láser que se alcanzó en dicho punto. Por su parte, en el fichero de la señal de vídeo, únicamente se registra el número del evento y el fotograma en el que se detectó el mismo.

1	65348	4356.489768	2.937729
2	136549	9103.175634	2.937729
3	233796	15586.244136	2.937729
4	314615	20974.12359	2.937729
5	399172	26611.200552	2.937729
6	480326	32021.413116	2.937729
7	555500	37032.963	2.937729
8	641268	42750.772488	2.932845
9	742603	49506.371597	2.937729
10	816343	54422.322437	2.937729
11	888921	59260.807385	2.937729
12	978948	65262.547367	2.937729
13	1053833	70254.830777	2.937729

1	97
2	380
3	767
4	1088
5	1424
6	1747
7	2046
8	2387
9	2789
10	3083
11	3371
12	3729
13	4027

Figura 4.1: Datos correspondientes a los artefactos detectados en la prueba del pez eléctrico (a la izquierda la señal electrofisiológica y a la derecha la señal de vídeo).

Según los datos recogidos en el fichero de la figura 4.1, la luz del puntero láser debería estar presente en el vídeo por primera vez en el fotograma 97 (que se aprecia levemente en la figura 4.2, ya que está comenzando a encenderse). En esta misma figura se puede observar también como se muestra la luz verde del puntero láser en los fotogramas 1088, 2046, 3083 y 4027.



Figura 4.2: Ejemplos de fotogramas en los que se detectó la luz del láser (prueba del pez eléctrico).

Por otro lado, en la imagen 4.3 podemos observar que efectivamente, los cuatro primeros artefactos detectados en el canal en el que se recoge el voltaje del puntero láser, se producen en los tiempos indicados en el fichero de la figura 4.1 (en la imagen se aprecia el valor aproximado correspondiente a la cota inferior de cada artefacto).

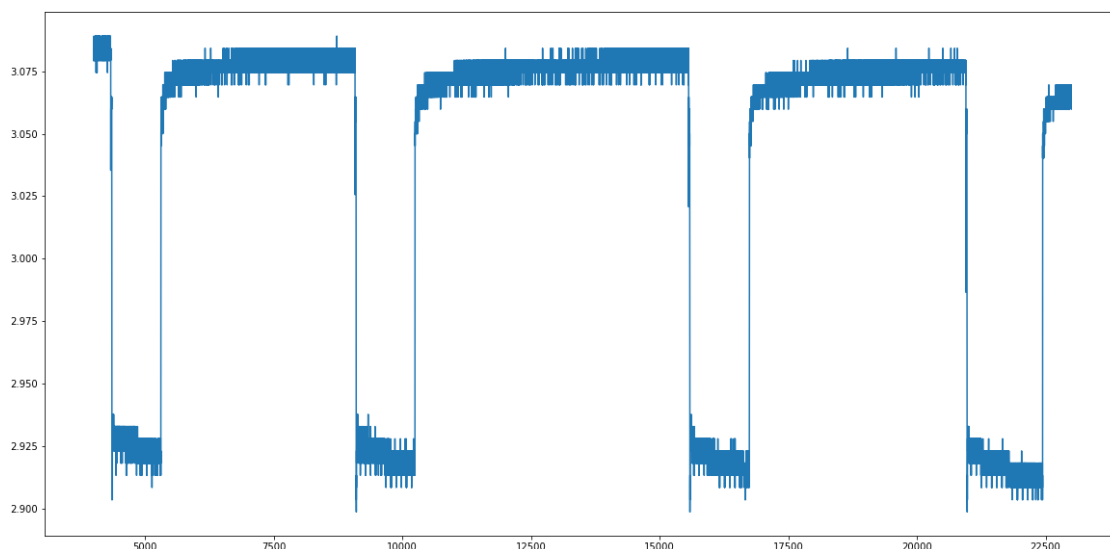


Figura 4.3: Primeros eventos de la señal que recoge el voltaje del láser (prueba del pez eléctrico).

Estas comprobaciones se realizaron con todas las muestras y fotogramas que se recogieron en los ficheros anteriores, y se corroboró que en todos ellos aparecía el artefacto.

#### 4.1.2. Análisis de la sincronización

Primeramente, vamos a mostrar que la sincronización se realiza de forma correcta. Originalmente, la señal de vídeo se empezó a registrar ligeramente más tarde que la señal electrofisiológica. Por ello, al reproducir ambas señales simultáneamente, sin llevar a cabo la sincronización, es de esperar que el artefacto se produzca antes en el vídeo que en la señal electrofisiológica, y esto se observa en la imagen 4.4. Esto se debe a que el tiempo transcurrido entre el inicio de la grabación y el momento en el que apareció el artefacto es menor que en el de la señal electrofisiológica, que comenzó a adquirirse antes.

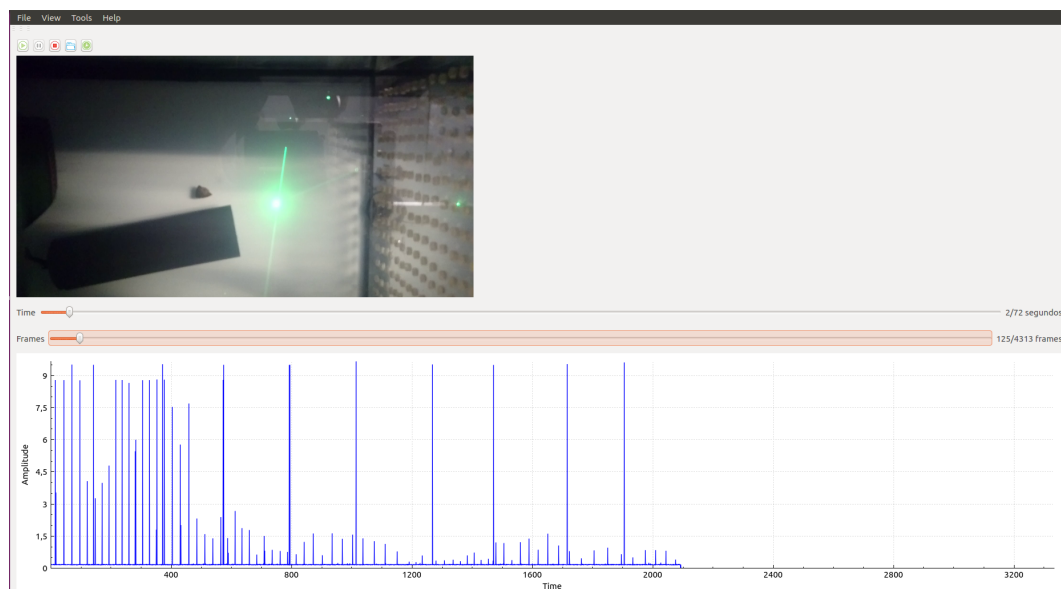


Figura 4.4: Primer artefacto visualizado en el vídeo, pero no en la señal electrofisiológica (prueba del pez eléctrico).

Sin embargo, al realizar la sincronización automática, se puede observar que se muestra el primer artefacto simultáneamente en la señal de vídeo y en la señal electrofisiológica (ver figura 4.5). La línea roja en la señal electrofisiológica indica la duración del artefacto, ya que en la aplicación no se visualiza la señal del voltaje del láser. Para que se vea más claro, se ha añadido debajo de la captura de la aplicación, la serie temporal de la señal del voltaje de la batería del puntero láser para comprobar que efectivamente aparece el artefacto al mismo tiempo (ver figura 4.6).

Si continuamos con la reproducción, se observa que, efectivamente, todos los artefactos aparecen simultáneamente en ambas señales. En la figura 4.7 se visualiza el último artefacto sincronizado. Además, también se puede apreciar un aumento considerable de la actividad eléctrica del pez, ya que en ese momento se está moviendo rápidamente.

Aunque visualmente la sincronización parece correcta, debemos estudiar la precisión de la sincronización, calculando el error en los tiempos en los que se detectaron los artefactos en ambas señales. Por un lado, en la señal de vídeo el error cometido por cada fotograma mostrado es de  $\pm \frac{1}{60}$  segundos, ya que la frecuencia de adquisición es de 60 fotogramas por segundo. Por otro lado, la señal electrofisiológica presenta una frecuencia de 15000 muestras por segundo, lo que lleva a cometer un error de  $\pm \frac{1}{15000}$  segundos. Para hacer más fácil en análisis, se mostrarán los distintos resultados en forma de tabla.

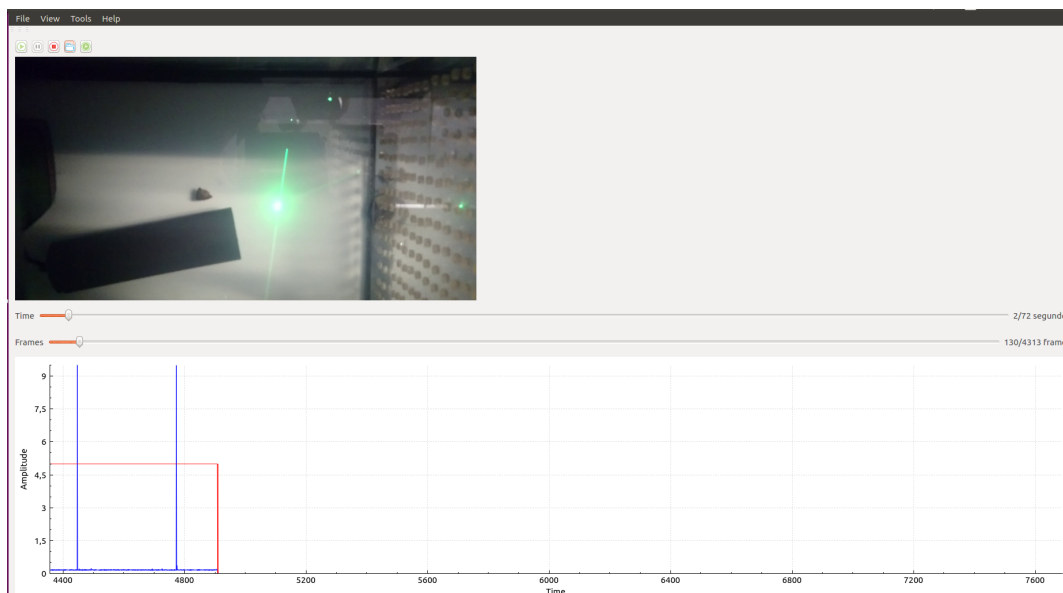


Figura 4.5: Primer artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba del pez eléctrico).

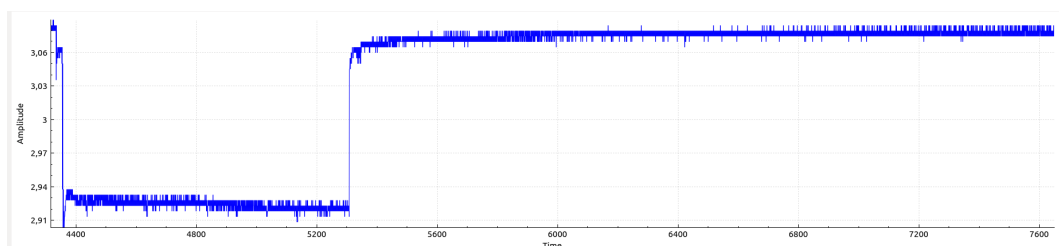


Figura 4.6: Primer artefacto visualizado en la señal que registra el voltaje del láser (prueba del pez eléctrico).

En la tabla 4.1 se muestran los tiempos recogidos por ambas señales tras detectar los eventos indicados en la primera columna. La segunda columna refleja los fotogramas a los que corresponde el evento, y la tercera al tiempo, en milisegundos, en el que la señal electrofisiológica detectó dicho evento. Para hacer más fácil los cálculos que vienen a continuación, en esta tabla también se muestran estos tiempos tomando como referencia de tiempo inicial ( $t=0$ ) el momento en el que sucedió el primer evento, ya que es el instante en el que se considera que comienza la reproducción simultánea tras la sincronización.

Mientras monitorizamos la señal eléctrica generada por el pez, nos gustaría saber cómo se está comportando en cada momento en el vídeo, y correlacionarlo con esa actividad eléctrica. Para ello, debemos seleccionar un instante de tiempo en la señal electrofisiológica y hallar el fotograma al que corresponde en la señal de vídeo. El procedimiento a seguir es el siguiente. Tomando un instante de tiempo en milisegundos, esta cantidad se debe dividir entre el tiempo que tarda en reproducirse un fotograma, que en esta prueba son  $\frac{1000}{60}$  milisegundos, con lo que hallaríamos el número de fotogramas que se han reproducido durante ese tiempo. Dado que tomamos como referencia de tiempo 0 el momento en el que se adquirió el primer evento, utilizamos para los cálculos los tiempos reflejados en la cuarta columna de la tabla anterior, y tras conocer el fotograma asociado, sumaremos 97 ya que es el fotograma de origen de la señal de vídeo. A continuación se detallan los cálculos realizados para el evento número 2 con el fin de servir como ejemplo.

Tenemos que el segundo evento se produjo en el instante 9103.175634 (en milisegundos),



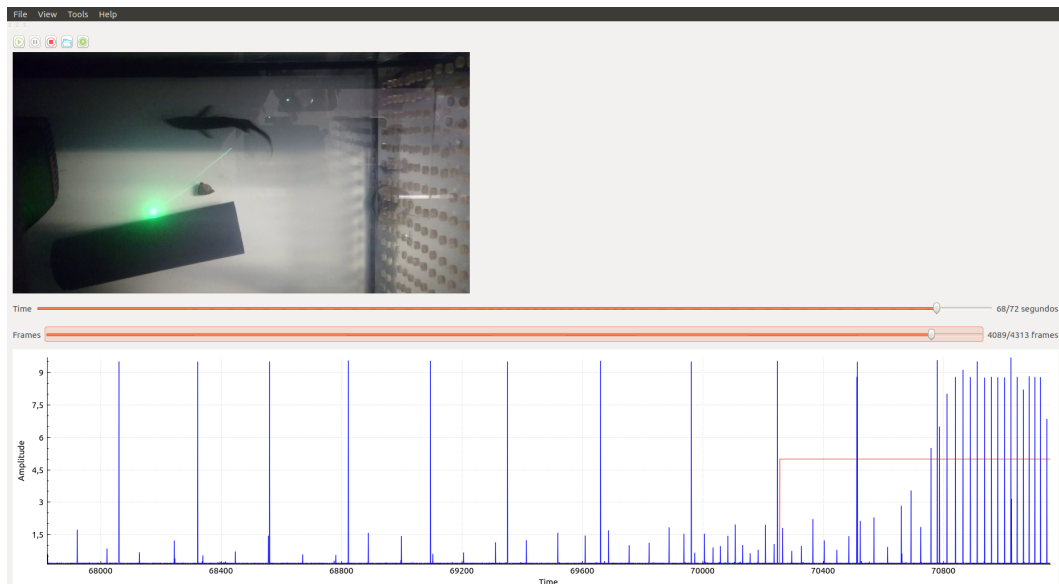


Figura 4.7: Último artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba del pez eléctrico).

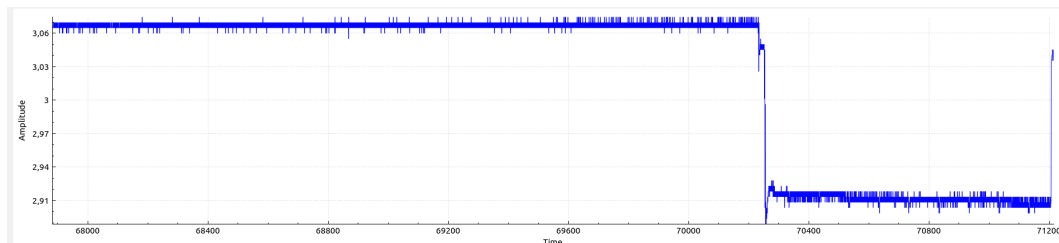


Figura 4.8: Último artefacto visualizado en la señal que registra el voltaje del láser (prueba del pez eléctrico).

pero suponiendo como origen de la señal el instante 4356.489768, cuando se produjo el primer evento, el tiempo que utilizamos es la diferencia entre estos valores, lo que nos da 4746.685866 milisegundos. Por tanto, los fotogramas reproducidos durante este tiempo aparecen como resultado del cociente  $\frac{4746.685866}{\frac{1000}{60}}$ , con lo que obtenemos aproximadamente 285 fotogramas (hemos redondeado la cantidad para conseguir un número entero, ya que no podemos desplazarnos a una porción de fotograma). A esta cantidad le sumamos 97, ya que es el fotograma en el que se detectó el primer evento, por lo que finalmente el fotograma correspondiente que buscamos es el 382. Este resultado y los obtenidos para el resto de eventos están recogidos en la tabla 4.2.

Por otro lado, también querríamos conocer qué actividad eléctrica se relaciona con un determinado comportamiento del pez que observamos en el vídeo. Para ello, dado un fotograma debemos obtener el instante de tiempo correspondiente en la señal electrofisiológica. El procedimiento es muy parecido al del caso anterior, pero en la otra dirección, por lo que pasamos directamente a explicar el proceso con un ejemplo.

Si tomamos el evento número 2, que se recogió en el fotograma 380, debemos usar la diferencia entre esta cantidad y 97, que es el fotograma que tomamos como origen de tiempo 0, con lo que obtenemos el valor 283. Esta cantidad la debemos multiplicar por el número de muestras que aparecen en un fotograma, que en este caso son 250, como ya se calculó anteriormente (ver fórmula 3.1), con lo que obtenemos 70750 muestras. Ahora tenemos que hallar el tiempo transcurrido tras reproducirse estas muestras, lo que resulta de la operación  $\frac{70750}{15000} \times 1000$  (la multiplicación por 1000 es para pasar a milisegundos), con lo que obtenemos 4716.66667

$Numero_{evento}$	$Frame_{video}$	$Tiempo_{electro}(ms)$	$Frame_{video}^{refo}$	$Tiempo_{electro}^{refo}(ms)$
1	97	4356.489768	0	0.0
2	380	9103.175634	283	4746.685866
3	767	15586.244136	670	11229.75436
4	1088	20974.12359	991	16617.63382
5	1424	26611.200552	1327	22254.71078
6	1747	32021.413116	1650	27664.92334
7	2046	37032.963	1949	32676.47323
8	2387	42750.772488	2290	38394.28271
9	2789	49506.371597	2692	45149.88182
10	3083	54422.322437	2986	50065.83266
11	3371	59260.807385	3274	54904.31761
12	3729	65262.547367	3632	60906.05759
13	4027	70254.830777	3930	65898.341

Cuadro 4.1: Tiempos originales y relativos al tiempo del evento de la sincronización (prueba del pez eléctrico).

$Numero_{evento}$	$Tiempo_{electro}(ms)$	$Frame_{video}^{refelectro}$	$Error_{frame}$
1	4356.489768	97	0
2	9103.175634	382	2
3	15586.244136	771	4
4	20974.12359	1094	6
5	26611.200552	1432	8
6	32021.413116	1757	10
7	37032.963	2058	12
8	42750.772488	2401	14
9	49506.371597	2806	17
10	54422.322437	3101	18
11	59260.807385	3391	20
12	65262.547367	3751	22
13	70254.830777	4051	24

Cuadro 4.2: Fotogramas asociados a tiempos de la señal electrofisiológica de los eventos (prueba del pez eléctrico).

milisegundos. Finalmente a este valor le sumamos 4356.489768, ya que es el instante que tomamos como origen de la señal electrofisiológica. Por tanto, el resultado final es 9073.156435. Este resultado y los obtenidos con el resto de eventos se reflejan en la tabla 4.3.

$Numero_{evento}$	$Frame_{video}$	$Tiempo_{electro}^{ref_{video}}(ms)$	$Error_{tiempo}(ms)$
1	97	4356.489768	0.0
2	380	9073.156435	30.019199
3	767	15523.156435	63.087701
4	1088	20873.156435	100.967155
5	1424	26473.156435	138.044117
6	1747	31856.489768	164.923348
7	2046	36839.823101	193.139899
8	2387	42523.156435	227.616053
9	2789	49223.156435	283.215162
10	3083	54123.156435	299.166002
11	3371	58923.156435	337.650950
12	3729	64889.823101	372.724266
13	4027	69856.489768	398.341009

Cuadro 4.3: Tiempos asociados a fotogramas de la señal de vídeo de los eventos (prueba del pez eléctrico).

Podemos observar, que en las tablas 4.2 y 4.3 se ha añadido una última columna que indica la diferencia entre el valor estimado en cada caso, y el valor original mostrado en la tabla 4.1. Se puede apreciar que el error va aumentando conforme aparecen nuevos eventos, es decir, el error se acumula a lo largo del tiempo. Esto se debe a que cuando igualamos las referencias temporales del primer evento recogidas por cada una de las señales, podríamos estar cometiendo un error de hasta 16 milisegundos aproximadamente ( $1/60$  segundos), ya que estamos colocando la muestra de la serie temporal que detectó el evento al inicio del fotograma, pero esta podría estar situada en realidad en cualquiera de las otras 249 posiciones (recordamos que se reproducen 250 muestras por cada fotograma). Si para el siguiente evento, la muestra que lo detectó en la señal electrofisiológica se encuentra en la posición 250 del fotograma de detección del evento, estaríamos cometiendo un error de 33 ms ( $2/60$  segundos), y así sucesivamente con el resto de eventos, por lo que se acumula el error.

Por este motivo, es recomendable crear varios eventos de sincronización e iniciar la reproducción simultánea a partir del evento más cercano al momento que queremos visualizar en ambas señales, de manera que el error cometido sea lo más pequeño posible.

## **4.2. Prueba de sincronización con una neurona electrónica**

---

En esta prueba se grabó en vídeo la actividad de una neurona electrónica visualizada en un osciloscopio durante 1 minuto y 24 segundos. En el entorno de grabación se añadió también un cronómetro con precisión hasta las centésimas de segundo, con el fin de comprobar que la grabación del vídeo se mantenía uniforme a su frecuencia de muestreo (que en este caso eran 60 fotogramas por segundo), y descartar la posibilidad de que se estuviera llevando a cabo una interpolación en la reproducción de los fotogramas. La luz del láser que generaba el artefacto en el vídeo aparece señalizando el osciloscopio.

Para registrar simultáneamente la señal de la neurona y los artefactos en la señal eléctrica derivados de encender el puntero láser, se conectó también la batería de éste al osciloscopio, reco-

giendo la diferencia de potencial en un segundo canal. En este caso, la frecuencia de adquisición de la señal de la neurona era de 10000 muestras por segundo, por lo que aplicando la fórmula 3.1 obtenemos que se debían reproducir 166 muestras de la señal eléctrica por cada fotograma reproducido en el vídeo. Como en el experimento anterior, el artefacto se generó hasta en 13 ocasiones.

#### 4.2.1. Validación de la detección de artefactos

Tal y como se ha explicado anteriormente, tras detectar los distintos artefactos en las dos señales la aplicación registraba la temporalidad de estos, apareciendo expresada en milisegundos en el caso de la señal eléctrica y en fotogramas en la señal de vídeo. En la figura 4.9 se observa el contenido de los ficheros que recogen la información citada, obtenida al realizar este experimento (en el apartado 4.1.1 se explicó cómo estaba organizada esta información).

1	159832	1939672.50066	2.621347	1	480
2	186604	1942349.70066	2.619516	2	639
3	218326	1945521.90066	2.622568	3	828
4	246132	1948302.50067	2.622568	4	994
5	282167	1951906.00067	2.623178	5	1209
6	319770	1955666.30067	2.620737	6	1433
7	351172	1958806.50068	2.622263	7	1620
8	378334	1961522.70068	2.622568	8	1782
9	400986	1963787.90068	2.620737	9	1917
10	426833	1966372.60068	2.622568	10	2072
11	454919	1969181.20068	2.622263	11	2239
12	481249	1971814.20069	2.620432	12	2396
13	509197	1974609.00069	2.623178	13	2563

Figura 4.9: Datos correspondientes a los artefactos detectados en la prueba de la neurona electrónica (a la izquierda la señal electrofisiológica y a la derecha la señal de vídeo).

Al igual que se hizo cuando se explicó el experimento con el pez eléctrico, ahora procederemos a comprobar visualmente que la detección de los artefactos se registró en el momento adecuado. En la figura 4.10 se puede apreciar la luz del puntero láser proyectada sobre el osciloscopio en los fotogramas 480, 994, 1620, 2072 y 2563, tal y como se registró en el fichero externo correspondiente a la señal de vídeo (ver parte derecha de la figura 4.9).

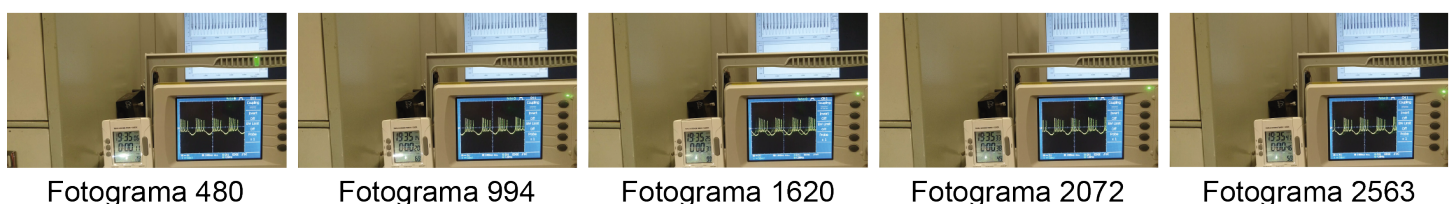


Figura 4.10: Ejemplos de fotogramas en los que se detectó la luz del láser (prueba de la neurona electrónica).

De la misma manera, en la figura 4.11 se aprecian los cuatro primeros artefactos detectados en la señal eléctrica tras activar el puntero láser, recogidos en el canal que registra la diferencia de potencial de su batería. Se puede observar que los tiempos asociados a la cota inferior de cada artefacto, coinciden aproximadamente con los anotados en el fichero externo correspondiente a la señal eléctrica (ver parte izquierda de la figura 4.9).

Todas estas comprobaciones se llevaron a cabo para todos los eventos detectados, tanto en los fotogramas de la señal de vídeo como para las muestras de la serie temporal de la señal eléctrica y todas ellas dieron los resultados deseados.

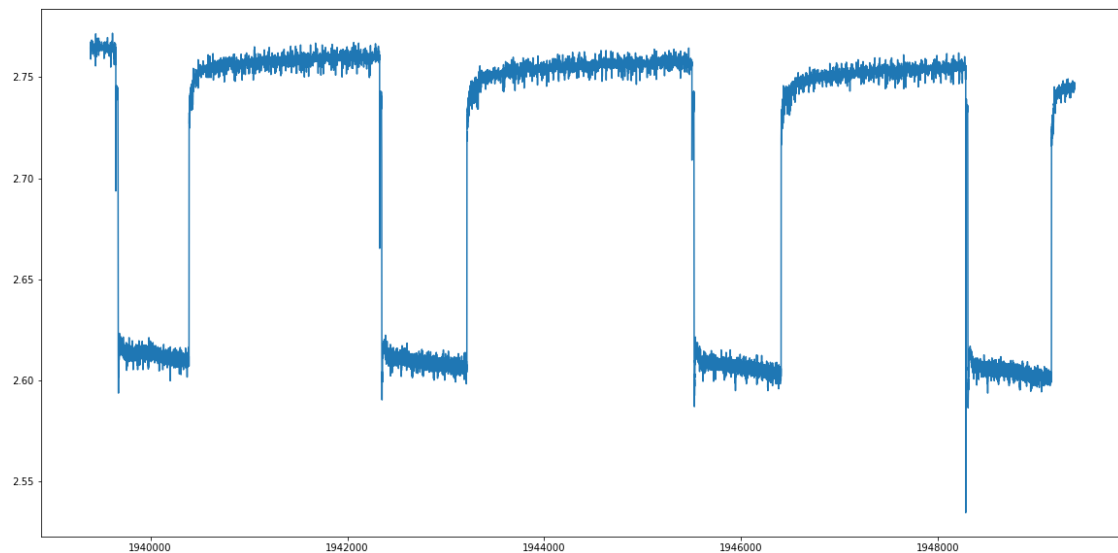


Figura 4.11: Primeros eventos de la señal que recoge el voltaje del láser (prueba de la neurona electrónica).

#### 4.2.2. Análisis de la sincronización

Antes de nada, vamos a comprobar, como ya adelantamos al comienzo de este apartado, que la señal de vídeo grababa de manera uniforme a lo largo del tiempo. Para ello, vamos a comprobar que entre el fotograma número 1 y el 600, han transcurrido 10 segundos, ya que la frecuencia de muestreo del vídeo es de 60 fotogramas por segundo ( $10s \times 60fps = 600\text{fotogramas}$ ). En la figura 4.12 se puede apreciar como el fotograma 1 el cronómetro indicaba 3 segundos, y en el fotograma 600 aparece el segundo 13, que es el valor que se esperaba obtener. También se observa una diferencia de 2 centésimas, la cuál se explica por el error cometido de  $1/60$  (debido a la frecuencia de muestreo). Por tanto, podemos afirmar que el reproductor de vídeo utilizado graba uniformemente, por lo que no existe interpolación de fotogramas.



Figura 4.12: Fotograma 1 (izquierda) y 600 (derecha) de la señal de vídeo.

A continuación se va a comprobar de manera visual que la sincronización de las señales se realiza correctamente. En el experimento, la señal de vídeo comenzó a registrarse unos segundos más tarde que la señal eléctrica, por lo que al reproducir ambas señales simultáneamente sin estar sincronizadas, se observa que el primer artefacto aparece antes en el vídeo (ver imagen 4.13), ya que ha transcurrido menos tiempo desde que se inició el registro de vídeo que desde que comenzó el registro de la neurona.

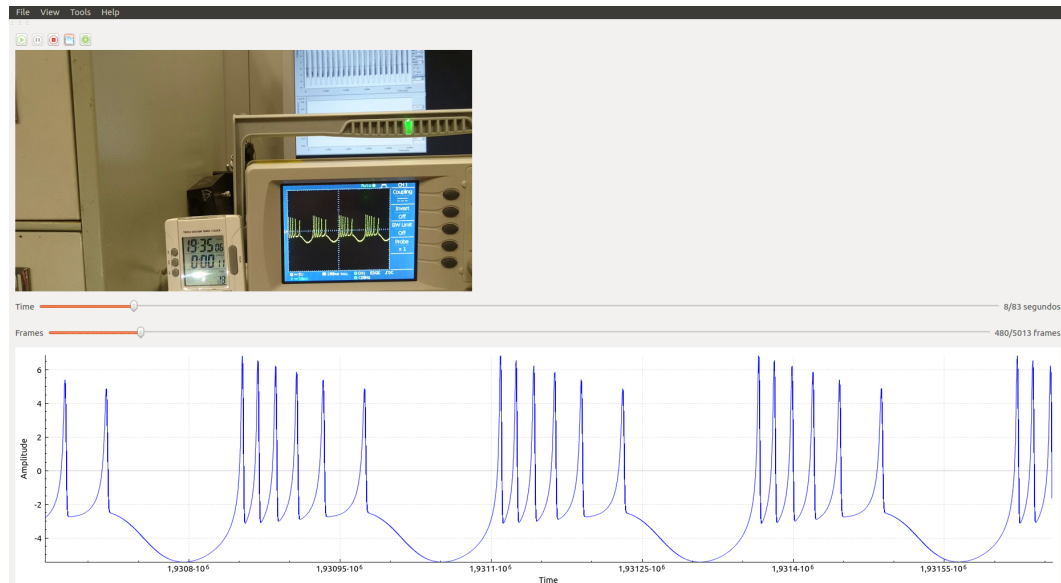


Figura 4.13: Primer artefacto visualizado en el vídeo, pero no en la señal electrofisiológica (prueba de la neurona electrónica).

Una vez realizada la sincronización automática, se puede observar en las imágenes 4.14 y 4.16 como aparecen en las dos señales simultáneamente el segundo y último artefacto detectado, respectivamente (se muestra el segundo artefacto en vez del primero, ya que éste era más difícil de capturar en una imagen). La línea vertical de color rojo indica el inicio del artefacto en la señal eléctrica. De la misma manera que hicimos con la prueba del pez eléctrico, debajo de estas imágenes se ha añadido la serie temporal de la señal que recogía la diferencia de potencial del puntero láser para verificar que el artefacto aparece al mismo tiempo que en el vídeo (se muestra sólo el inicio del artefacto en la parte derecha de la imagen).

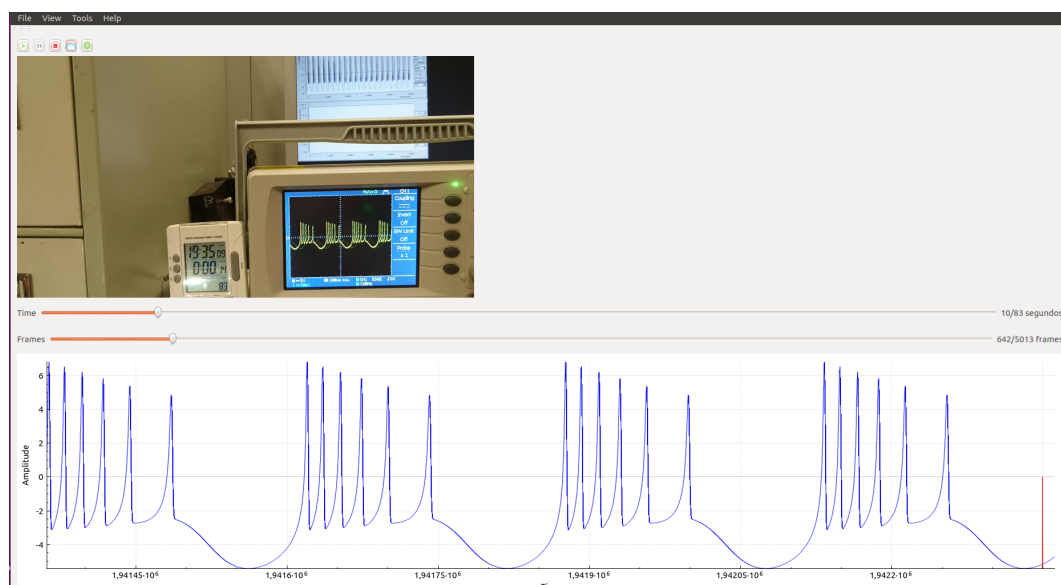


Figura 4.14: Segundo artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba de la neurona electrónica).

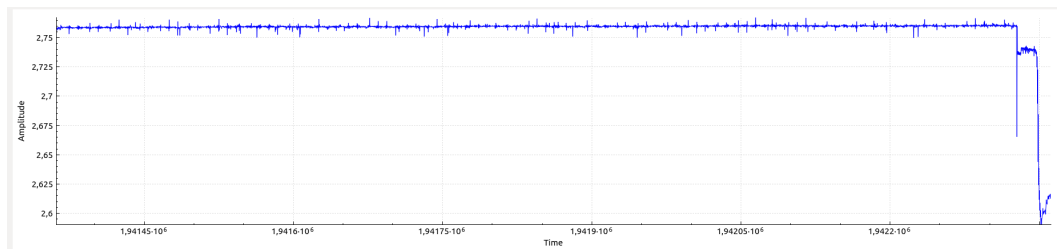


Figura 4.15: Segundo artefacto visualizado en la señal que registra el voltaje del láser (prueba de la neurona electrónica).

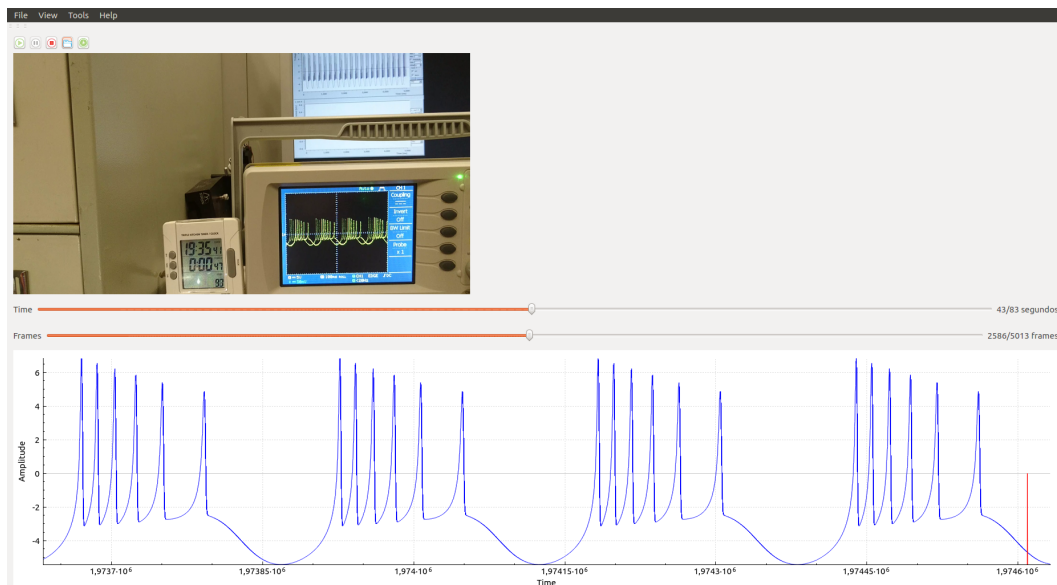


Figura 4.16: Último artefacto visualizado en el vídeo y en la señal electrofisiológica (prueba de la neurona electrónica).

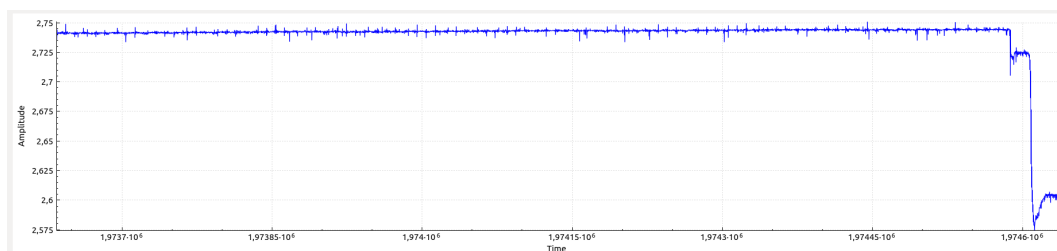


Figura 4.17: Último artefacto visualizado en la señal que registra el voltaje del láser (prueba de la neurona electrónica).

Sin embargo, se observa un ligero retraso en la señal electrofisiológica respecto al vídeo, el cual se debe al pintado de la gráfica. Otra razón es la diferencia entre las frecuencias de adquisición de ambas señales. Por estos motivos, se va a proceder a analizar la precisión de la sincronización, utilizando tablas similares a las expuestas en la prueba anterior.

Las siguientes tablas se han organizado de forma similar a las que se explicaron para la prueba del pez eléctrico, así como los cálculos realizados para completarlas han seguido el mismo procedimiento detallado para la misma (en caso de duda ver explicación de las tablas 4.1, 4.2 y 4.3). Hay que tener en cuenta, que en este caso la frecuencia de adquisición de la señal electrofisiológica es de 10000 muestras por segundo y no 15000 como ocurría en el caso anterior.



Observando los resultados obtenidos en la tabla 4.5, vemos que el error cometido se acumula de nuevo como en el caso anterior, sin embargo, en este caso el error parece algo menor. Esto se debe a que la frecuencia de muestreo de la señal electrofisiológica es más baja, por lo que la diferencia entre las frecuencias de adquisición de las dos señales también disminuye, lo que conlleva menos pérdida de precisión.

$Numero_{evento}$	$Frame_{video}$	$Tiempo_{electro}(ms)$	$Frame_{video}^{ref_0}$	$Tiempo_{electro}^{ref_0}(ms)$
1	480	1939672.50066	0	0.0
2	639	1942349.70066	159	2677.2
3	828	1945521.90066	348	5849.4
4	994	1948302.50067	514	8630.00001
5	1209	1951906.00067	729	12233.50001
6	1433	1955666.30067	953	15993.80001
7	1620	1958806.50068	1140	19134.00002
8	1782	1961522.70068	1302	21850.20002
9	1917	1963787.90068	1437	24115.40002
10	2072	1966372.60068	1592	26700.10002
11	2239	1969181.20068	1759	29508.70002
12	2396	1971814.20069	1916	32141.70003
13	2563	1974609.00069	2083	34936.50003

Cuadro 4.4: Tiempos originales y relativos al tiempo del evento de la sincronización (prueba de la neurona electrónica).

$Numero_{evento}$	$Tiempo_{electro}(ms)$	$Frame_{video}^{ref_{electro}}$	$Error_{frame}$
1	1939672.50066	480	0
2	1942349.70066	641	2
3	1945521.90066	831	3
4	1948302.50067	998	4
5	1951906.00067	1214	5
6	1955666.30067	1440	7
7	1958806.50068	1628	8
8	1961522.70068	1791	9
9	1963787.90068	1927	10
10	1966372.60068	2082	10
11	1969181.20068	2251	12
12	1971814.20069	2409	13
13	1974609.00069	2576	13

Cuadro 4.5: Fotogramas asociados a tiempos de la señal electrofisiológica de los eventos (prueba de la neurona electrónica).



$Numero_{evento}$	$Frame_{video}$	$Tiempo_{electro}^{ref_{video}}(ms)$	$Error_{tiempo}(ms)$
1	480	1939672.50066	0.0
2	639	1942311.90066	37.8
3	828	1945449.30066	72.6
4	994	1948204.90066	97.60001
5	1209	1951773.90066	132.10001
6	1433	1955492.30066	174.00001
7	1620	1958596.50066	210.00002
8	1782	1961285.70066	237.00002
9	1917	1963526.70066	261.20002
10	2072	1966099.70066	272.90002
11	2239	1968871.90066	309.30002
12	2396	1971478.10066	336.10003
13	2563	1974250.30066	358.70003

Cuadro 4.6: Tiempos asociados a fotogramas de la señal de vídeo de los eventos (prueba de la neurona electrónica).



# 5

## Conclusiones y trabajo futuro

En este proyecto se ha descrito una aplicación que permite llevar a cabo una monitorización simultánea y sincronización automática de una señal de comportamiento en vídeo y una señal electrofisiológica que recoge la actividad eléctrica emitida por una entidad biológica. Esta herramienta resulta de gran utilidad en distintos ámbitos de investigación, como son la neurociencia, la biomedicina o la neuroetología, ya que permite estudiar la correlación entre un determinado comportamiento y su correspondiente respuesta electrofisiológica, y viceversa.

A lo largo de este documento, se han ido comprobando uno a uno los objetivos que se esperaban alcanzar. La aplicación permite visualizar una señal de vídeo y otra electrofisiológica, llevando a cabo una adaptación de las distintas frecuencias de sampleo. También se ha comprobado que generando un artefacto simultáneo y visible en ambos registros (que en este caso se ha elegido la activación de un puntero láser), es posible llevar a cabo una sincronización automática de las señales tras detectar los eventos generados, ya que de esta manera se puede conseguir una referencia temporal común a las dos señales. Se ha visto también, que existe cierta pérdida de precisión en la sincronización, debida fundamentalmente a las diferentes frecuencias de muestreo de las señales, por lo que la sincronización será mejor cuanto más se parezcan estas frecuencias. Por otro lado, el estudio muestra que es útil utilizar varios artefactos de sincronización para aumentar la sincronización.

En cuanto al trabajo futuro, expondremos a continuación algunas de las más relevantes:

- **Detección de otro tipo de eventos:** Los eventos sonoros por ejemplo, pueden resultar fácilmente reconocibles tanto en una señal electrofisiológica como en una señal de vídeo. Así mismo, se podría también detectar distinto tipo de artefactos dentro de un mismo registro.
- **Implementación multiplataforma:** Actualmente, el prototipo de la aplicación únicamente está disponible para el sistema operativo Linux, con lo que esta mejora conlleva varias modificaciones, como por ejemplo el acceso al árbol de directorios a la hora de cargar los registros, ya que el formato puede ser distinto al de otros sistemas operativos. Esta mejora se podría realizar fácilmente para Windows, ya que tanto OpenCV como Qt presentan distribuciones para este sistema operativo.
- **Monitorización y sincronización de múltiples señales de vídeo y de electrofisiología:** Esta mejora permitiría desarrollar experimentos de mayor envergadura, ya que

se podrían visualizar distintas partes del entorno con distintas cámaras y diferentes zonas con actividad eléctrica de la entidad biológica con diversos electrodos.

- **Selección del evento de sincronización:** En esta aplicación, la sincronización se lleva a cabo a partir del primer evento generado. Sin embargo, si se pudiera elegir a partir de qué evento se quiere iniciar la visualización simultánea, se reduciría el error de precisión acumulado, provocado por la inevitable diferencia entre las frecuencias de muestreo de las señales (ya que a día de hoy, la señales eléctricas presenta una velocidad de transmisión mucho mayor que las señales de vídeo). Se podrían utilizar algoritmos de selección del mejor alineamiento de los eventos sincronizados [16].
- **Etiquetado de eventos:** Resultaría muy útil el hecho de poder etiquetar los eventos de especial interés, de manera que el investigador pudiera acceder a ellos en cualquier momento.

## Glosario de acrónimos

- **Electrofisiología:** Estudio de las propiedades eléctricas de células y tejidos biológicos. En neurociencia, se incluyen las medidas de la actividad eléctrica de neuronas, y particularmente actividad de potencial de acción. Registros a gran escala de señales eléctricas del sistema nervioso como la electroencefalografía (EEG), también se pueden clasificar como registros electrofisiológicos.
- **Neurociencia:** Campo unificado del conocimiento en la ciencia experimental que estudia la estructura, la función, el desarrollo de la bioquímica, la farmacología y la patología del sistema nervioso y de cómo sus diferentes elementos interactúan, dando lugar a las bases biológicas de la conducta.
- **Neuroetología:** Enfoque evolutivo y comparativo del estudio de la conducta animal y sus correlatos en el sistema nervioso.
- **Biomedicina:** Se relaciona con la práctica de la medicina, y aplica todos los principios de las ciencias naturales en la práctica clínica, mediante el estudio e investigación de los procesos fisiopatológicos.
- **Artefacto:** Distorsión visible en una señal. En este proyecto se han utilizado artefactos simultáneos generados en dos señales para llevar a cabo su sincronización.
- **EEG:** Electroencefalografía, medida neurofisiológica que se basa en el registro de la actividad bioeléctrica cerebral. Puede realizarse en distintos periodos: reposo, vigilia, o sueño.
- **GNB:** Grupo de Neurocomputación Científica
- **FPS:** Fotogramas por segundo
- **RGB:** Red, Green, Blue. Es el formato de color más utilizado.
- **HSV:** Hue, Saturation, Value. Es un formato de color que facilita el análisis de imágenes.
- **OpenCV:** Conjunto de librerías disponibles en C++ y Python especializadas en la gestión, análisis y procesamiento de imágenes en tiempo real.
- **Qt:** Biblioteca multiplataforma disponible para C++ y Python que se utiliza principalmente en la creación de interfaces de usuario, dado su amplio catálogo de elementos gráficos. También cabe destacar su gestión de ventanas y su mecanismo de comunicación entre objetos.



# Bibliografía

- [1] Gonalo Lopes, Niccolò Bonacchi, Joao Frazao, Joana P. Neto, Bassam V. Atallah, Sofia Soares, Luíjs Moreira, Sara Matias, Pavel M. Itskov, Patrícia A. Correia, Roberto E. Medina, Lorenza Calcaterra, Elena Dreosti, Joseph J. Paton, and Adam R. Kampff. Bonsai: an event-based framework for processing and controlling data streams. *Frontiers in Neuroinformatics*, 9:7, apr 2015.
- [2] Olivier Darbin, Otis Smart, and Thomas Wichmann. A non-invasive technique to monitor wakefulness during electrophysiologic recording experiments in primates. *Journal of Neuroscience Methods*, 177(2):448–451, 2009.
- [3] Carlos Muñiz, Francisco Rodríguez, Pablo Varona, Reynaldo D Pinto, Francisco B Rodríguez, and Pablo Varona. RTBiomanager: a software platform to expand the applications of real-time technology in neuroscience. *BMC Neuroscience*, 10(Suppl 1):P49, jul 2009.
- [4] Carlos Muñiz, Caroline G Forlim, Rafael T Guariento, Reynaldo D Pinto, Francisco B Rodríguez, and Pablo Varona. Online video tracking for activity-dependent stimulation in neuroethology. *BMC Neuroscience*, 12(1):P358, jul 2011.
- [5] Laszlo Grand, Sergiu Ftomov, and Igor Timofeev. Long-term synchronized electrophysiological and behavioral wireless monitoring of freely moving animals. *Journal of Neuroscience Methods*, 212(2):237–241, 2013.
- [6] Evan R Gedzelman and Suzette M LaRoche. Long-term video EEG monitoring for diagnosis of psychogenic nonepileptic seizures. *Neuropsychiatric disease and treatment*, 10:1979–86, 2014.
- [7] Jacobo Fernandez-Vargas, Hanns U. Pfaff, Francisco B. Rodríguez, and Pablo Varona. Assisted closed-loop optimization of SSVEP-BCI efficiency. *Frontiers in Neural Circuits*, 7:27, 2013.
- [8] Álvaro Morán García. UNIVERSIDAD AUTÓNOMA DE MADRID ESCUELA POLITÉCNICA SUPERIOR PROYECTO FIN DE CARRERA DISEÑO DE INTERFACES CEREBRO-MÁQUINA CONTROLADOS MEDIANTE REGISTROS DE EEG. 2015.
- [9] Manuel Marcos García. UNIVERSIDAD AUTÓNOMA DE MADRID PROYECTO FIN DE CARRERA DISEÑO Y CONTROL DE UN ROBOT MEDIANTE INTERFAZ CEREBRO-MÁQUINA USANDO LA TECNOLOGÍA SSVEP. 2015.
- [10] Pablo Chamorro, Carlos Muñiz, Rafael Levi, David Arroyo, Francisco B. Rodríguez, and Pablo Varona. Generalization of the dynamic clamp concept in neurophysiology and behavior. *PLoS ONE*, 7(7):e40887, jul 2012.
- [11] Víctor Hugo García García and Víctor Hugo. Desarrollo de una "toolbox" para condicionar el comportamiento de peces eléctricos, basado en la codificación de las señales eléctricas emitidas por éste y su comportamiento. 2014.

- [12] Caroline Garcia Forlim, Reynaldo Daniel Pinto, Pablo Varona, Francisco B. Rodríguez, P Varona, and P Varona. Delay-Dependent Response in Weakly Electric Fish under Closed-Loop Pulse Stimulation. *PLOS ONE*, 10(10):e0141007, oct 2015.
- [13] Benjamin Suter, Timothy O'Connor, Vijay Iyer, Leopoldo Petreanu, Bryan M Hooks, Taro Kiritani, Karel Svoboda, and Gordon M. G Shepherd. Ephus: multipurpose data acquisition software for neuroscience experiments. *Frontiers in Neural Circuits*, 4:100, 2010.
- [14] Patrick D Martin, Hisao Nishijo, and Taketoshi Ono. A combined electrophysiological and video data acquisition system using a single computer. *Journal of Neuroscience Methods*, 92(1-2):169–177, 1999.
- [15] Alicia García Sastre and Alicia. Monitorización simultánea de comportamiento y señalización biológica eléctrica. 2016.
- [16] Luis F. Lago-Fernández, Attila Szücs, and Pablo Varona. Determining Burst Firing Time Distributions from Multiple Spike Trains. *Neural Computation*, 21(4):973–990, apr 2009.